Location+

By Jared Maeyama

As baseball evolves there is more and more data on the individual pitches each player throws. Now instead of just the mph that a pitcher throws a pitch at, there are sensors that read the spin rate of the ball and the axis that it spins on. Take Twins pitcher Jhoan Duran. We can see that his 4 seam fastball doesn't just average 101.8 mph, but we also know that its 2103 rmps combined with a 97% spin efficiency makes it break 10.9 inches horizontally and drop 11.9 inches vertically. These are very quantifiable numbers to see how well a pitch moves and the velocity it comes at a hitter. I wanted to have a quantifiable way to evaluate players ability to locate their pitches

A little background with all the data in the MLB. There are stats that evaluate how good a pitcher's pitch, stuff, is a vacuum, stuff+, but there is less data outside of rudimentary walk and waste pitch%, strike%. The few control metrics available less so utilize the influx of information that is now available in mlb through statcast, sensors that track everything on the field. These metrics are also not normalized like other stats. Stats ending in + are normalized so 100 is average and 110 is 10 percent better than league average and 90 is 10 percent worse than league average.

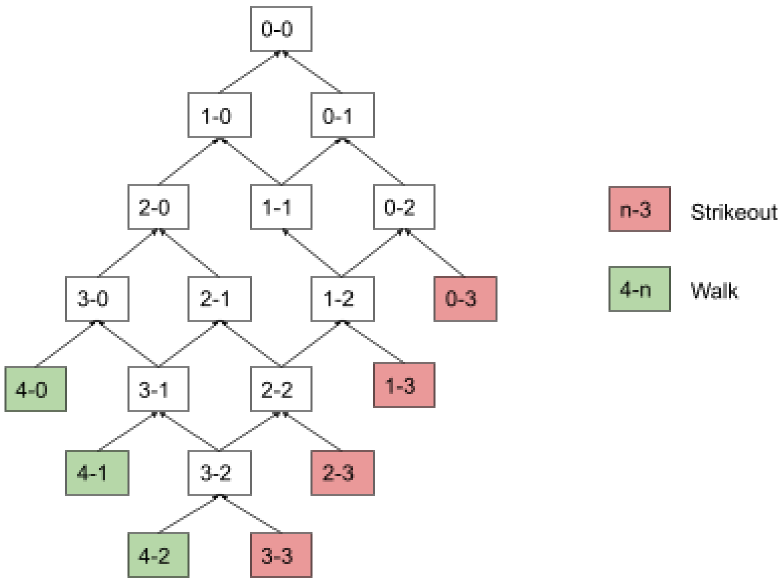
I would like to evaluate MLB pitcher locations. I'd use a data set of the mlb pitches of the 2022 season. I would evaluate just the player's location, taking out how good their pitch velocity, spin is. Given the pitch count and type of pitch, I'd make buckets that would have the impact of pitches in that location whether it is a batted ball or a change in the count. That would be normalized against the average impact of a pitch in the count. At the end, it would have how a pitchers pitch location affected the outcome of the at bat while controlling for the inherent quality of the pitch.

The thinking behind my project was seeing how much the location of a pitcher's pitch would affect their outcomes. I would need to control for the count and pitch type being thrown as a pitch thrown just outside of the zone is more effective when the count is 0-2 as opposed to 3-1. The reason for controlling pitch type is, generally breaking balls are more effective when lower in the zone but fastballs can be more effective when thrown higher in the zone. When a pitch is thrown there are 3 main outcomes, it's a ball, it's a strike or it's a batted ball. If it's a ball or strike then the count will advance and if it's hit then it's the end of the at bat. I used woba to evaluate pitchers as it property weights singles, doubles, triples, home runs hit by pitches and walks. I found the probability of each outcome and used the respective corresponding wobas and multiplied them together so x% of the time there will be a woba of the affected outcome. For the batted balls I just took the average of all batted balls. To get the expected woba for the batted ball outcome. Giving me this equation for the expected woba

$$xwoba\,[b - s] = \frac{(\#of\ pitches\ resulting\ in\ Ball)}{total\ pitches}(xwoba[(b + 1) - s]) + \frac{(\#of\ pitches\ resulting\ in\ Strike)}{total\ pitches}$$

$$(xwoba[b - (s + 1)]) + \frac{(\#of\ pitches\ resulting\ in\ Batted\ Ball)}{total\ pitches}(average\ of\ batted\ balls)$$

Since the equation is recursive then it needs a base case. The base cases are when a batter strikes out and when a batter walks. When a batter strikes out then their woba is 0 and based off of fangraphs formula then a walk is equivalent to woba .689. So whenever there are 3 strikes or there are 4 balls then there is a known woba then from working back I can calculate all of the woba given the count. An interesting note is that when a pitch is fouled off when the batter has two strikes then the count stays where it's at and it is counted as a strike. I had to remove pitches that resulted in foul balls in two strike counts.



I also needed to calculate the xwoba for a given pitch in a count using the same formula. Now to control location, I used the same strategy but I just limited the type of pitch and to all pitches in a 10 cm x 10 cm box.

$$xwoba\ for\ pitch = \frac{(\#of\ pitches\ resulting\ in\ Ball)}{total\ pitches}(xwoba[(b+1)-s]) + \frac{(\#of\ pitches\ resulting\ in\ Strike)}{total\ pitches}(xwoba[b-(s+1)]) + \frac{(\#of\ pitches\ resulting\ in\ Batted\ Ball)}{total\ pitches}(average\ of\ batted\ balls)$$

To get the quality of the pitches location is normalized it by

$$adjusted\ location = xwoba\ for\ pitch/woba\ for\ count\ and\ pitch\ type * 100$$

After average the adjusted location for all the pitches a player threw then you'd get their weighted location

First, I wanted to download all the pitching data for the 2022 season. Lucky Baseball Savant has that. Using their search function I can see all the data that savant has on each pitch. It has so much. It has the pitcher and hitter, who all the players in the field are, runners on base but not just who was playing but information of the pitch thrown. Like the xyz release position of the pitch and the z,x position of the pitch when it crosses the plate and more. The only problem is when I tried to download all the pitches of the 2022 season the file would be too big and the server would evenly give up downloading, but I could download individually by pitcher. There's a lot of pitchers to go through so I wanted to have code to download for me. Clicking on the link to try to find the hyper link didn't work. After looking around the code for the link, I was able to find a request to another page and luckily for me it was able to use this to download the individual files. I just needed the player ids for each player. The player ids are from Baseball Savant so I would need to get the id’s of all pitchers who are pitching in 2022 and use them to modify the url I found to get the resulting players data on the players pitching of the season. I got the ids by using the request library to get the html data from the list of pitchers. After the request, I used a regular expression to extract the ids and names and stored them into a dictionary with name as the term and the player id as the definition.

When I started to export the files to a folder. I noticed one of the players that I knew pitched a lot during the 2022 season had a really small file size. The file size reflects the amount of pitches that the pitcher threw in the 2022 season. When examining the file it only had data from 3 games that Luis Castillo pitched. When I looked up one of the listed games it said that he played for a different team that I remember him playing for. I realized that there were two Luis

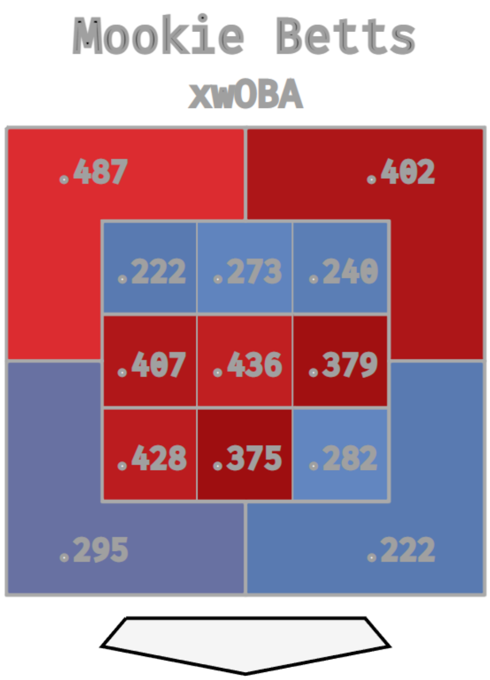
Castillos that pitched in 2022. One being the all star starting pitcher that I remember, and another who pitched 3 innings and is now playing in Japan. By storing that data into a dictionary based on the name of the player, it was overriding the previous pitcher's id and so it was not downloading their information. So I had to rerun the regex to just extract the player id. After using the modified urls I had to truncate the data so it would be written properly into the csv

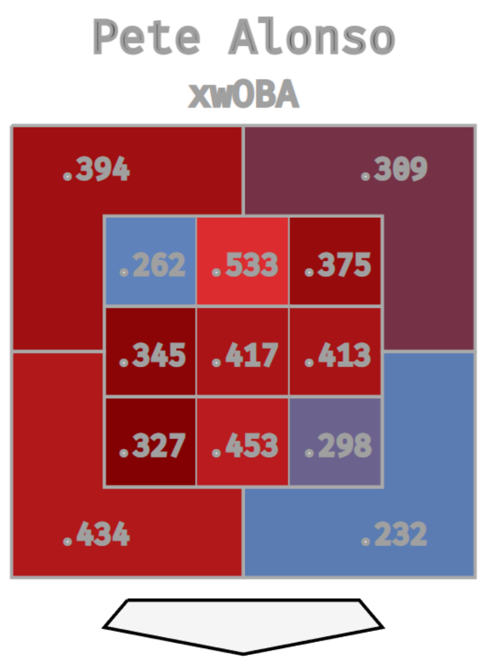
After getting all the information into files, I read in all the files and combined them into one all encompassing sheet to use. I made copies of the master sheet when making calculations for the different wobas. I stored the woba given count and pitch into a dictionary so they would be labeled. Then I iterated over all the rows taking all pitches of the same pitch type in a given count and that were +- 5 cm in either direction to get an expected woba for the pitch. I stored this in a dictionary where the key was the player's id and definition was a list of all the players pitches normalized. After averaging out the list then I have the player's location+.

Interesting outcome is that all of the players with the highest location+ score were position players. Position players are players that usually just hit and play defense. Teams will put them into pitch in blowout games to save the bullpen from more use. They tend to only pitch in at most a few games a season resulting in a small sample size for them making their location score very prone to outliers. Out of actual pitchers I found that Kevin Gausman had the highest score. Which was interesting to me since he throws a splitter, a pitch that not many other players throw.

Some drawbacks of the formula I used are that it over simplifies some pitches and doesn't take into account if the pitcher is using one pitch to set up other pitchers. There is a term called tunneling where two pitches that are thrown will both look like they are headed to the same location but one will break differently and end up in a different location. For example, a fastball up in the zone will tunnel with a curveball thrown in the bottom part of the zone. Fastballs typically end in the location where it looks like they will and where a curveball will end up substantially lower in the zone.

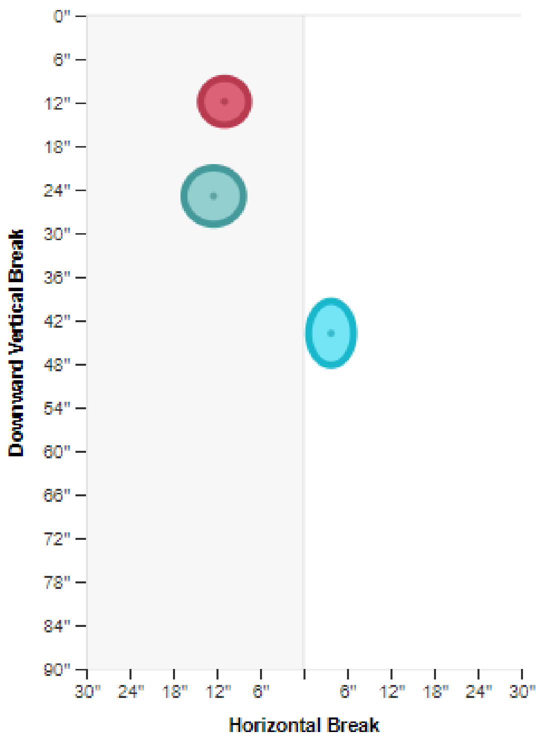
Also the model does not distinguish with the individual player's weakness at the plate as for example some players might crush inside pitches and another might struggle with inside pitches.

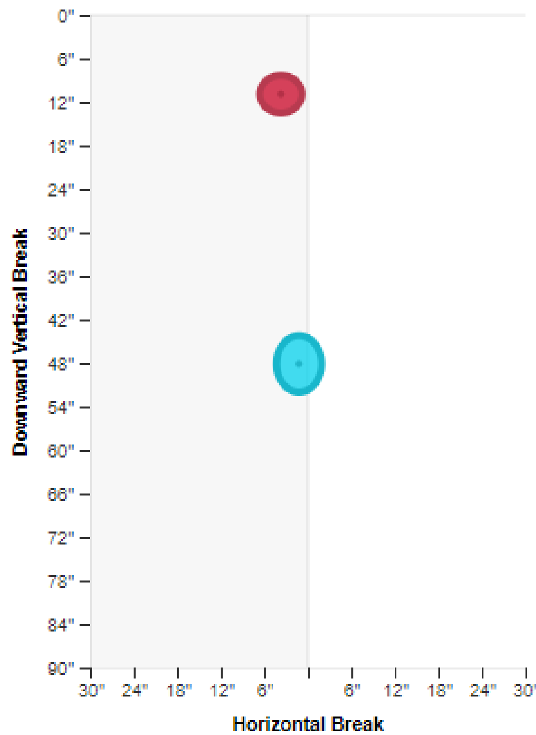




For example looking at the pitch middle up against Mookie Betts it's a pretty good pitch with a .273 xwoba which relative to the other pitch location is a very good outcome. Where looking at Pete Alonso the middle up pitch results in the highest xwoba of his chart. So a pitch middle up to Mookie Betts is a good pitch while a pitch middle up to Pete Alonso is the worst pitch location you can throw. So the algorithm does not account for how teams will attack hitters. Also if a team has a certain way that they want to attack a hitter it does not account for that if their strategy is against the standard pitch zones.

Also some pitches perform better based on the movement in different parts of the zone. Take the aforementioned Jhoan Duran and James Karichak.





With Jhoan Durans fastball in red on the left and James Karichak's fastball on the right there is about a 7 inch difference in horizontal movement between the two. Meaning tha duran's pitch is more likely to play better when thrown horizontally just off of the plate. Therefore optimal locations for Duran's fastball would differ then Karichaks.

A Lot of the problems would be solved with a bigger sample size to allow tighter restrictions so the pitches to control for more variables.

Works cited

"Guts!: Fangraphs Baseball." *Guts! | FanGraphs Baseball*, www.fangraphs.com/guts.aspx?type=cn. Accessed 11 June 2023.

"Jhoan Duran Statcast, Visuals & Advanced Metrics: Mlb.Com." *Baseballsavant.Com*, baseballsavant.mlb.com/savant-player/jhoan-duran-661395?stats=statcast-r-pitching-mlb. Accessed 11 June 2023.

"Mookie Betts Statcast, Visuals & Advanced Metrics: Mlb.Com." *Baseballsavant.Com*, baseballsavant.mlb.com/savant-player/mookie-betts-605141?stats=statcast-r-zones-mlb. Accessed 11 June 2023.

"Pete Alonso Statcast, Visuals & Advanced Metrics: Mlb.Com." *Baseballsavant.Com*, baseballsavant.mlb.com/savant-player/pete-alonso-624413?stats=statcast-r-hitting-mlb. Accessed 11 June 2023.

"Pitch Arsenal." *Baseballsavant*, baseballsavant.mlb.com/pitchmix#675916_2023. Accessed 11 June 2023.

"Pitch Arsenal." *Baseballsavant.Com*, baseballsavant.mlb.com/pitchmix#661395_2023. Accessed 11 June 2023.

"Statcast Custom Leaderboards." *Baseballsavant.Com*, baseballsavant.mlb.com/leaderboard/custom?year=2022&type=pitcher&filter=&sort=4&sortDir=asc&min=q&selections=xba%2Cxslg%2Cxwoba%2Cxobp%2Cxiso%2Cexit_velocity_avg%2

Claunch_angle_avg%2Cbarrel_batted_rate%2C&chart=false&x=xba&y=xba&r=no&chartType=beeswarm. Accessed 11 June 2023.

"Statcast Search." *Baseballsavant.Com*, baseballsavant.mlb.com/statcast_search?hfPTM=&hfPT=&hfAB=&hfGT=R%7C&hfPR=&hfZ=&hfStadium=&hfBBL=&hfNewZones=&hfPull=&hfC=&hfSea=2022%7C&hfSit=&player_type=pitcher&hfOuts=&hfOpponent=&pitcher_throws=&batter_stands=&hfSA=&game_date_gt=&game_date_lt=&hfMo=&hfTeam=&home_road=&hfRO=&position=&hfInfield=&hfOutfield=&hfInn=&hfBBT=&hfFlag=&metric_1=&group_by=name&min_pitches=0&min_results=0&min_pas=0&sort_col=pitches&player_event_sort=api_p_release_speed&sort_order=desc#results. Accessed 11 June 2023.