Strategic Use of Pitch Challenging in Baseball

Wyatt Johnston

September 7, 2026

Key Findings

I find that, as a default approach, players should challenge a call if they are at least **50% confident** they will succeed. This simple rule outperforms strategies that require high confidence or follow other heuristics.
In head-to-head comparisons, this yields a **2.0%-2.5% per-game win probability advantage**, corresponding to roughly **3-4 additional wins over a full season**.
For a modest additional edge (approximately **0.4%**), the optimal strategy incorporates the *count*: challenge conservatively early in the count, aggressively late, and at 50% otherwise. Rules based on inning, outs, or other game context perform substantially worse.

Summary

Major League Baseball is giving players the ability to challenge ball/strike calls beginning in the 2026 season. Here are the details the league has released about the new rule:

1. Each team gets two challenges and retains them if the challenge is successful.
2. Challenges can only be initiated by a pitcher, catcher, or batter, and the request must come immediately after the pitch.
3. To signal a challenge, the pitcher, catcher, or batter taps their hat or helmet to notify the umpire.
4. No help from the dugout or other players on the field is allowed.
5. In each extra inning, a team is awarded a challenge if it has none remaining when the inning begins.

This rule raises a practical question: *when should you challenge a pitch?* A closely related question is: *what is the value of a challenge?* I answer both using a stochastic model and available MLB pitch data. For one, the answers to these questions turn out to be fairly interesting. Beyond that, they could be of real consequence. Over the course of a season, teams that use their challenges better than others can expect to win a handful (perhaps 3 to 4) more games.

1 The Need For A Strategy

You might think that the question "When should you challenge a pitch?" is trivial because the answer is obvious: you should challenge when the pitch was called wrong. Of course, that would be the best strategy, but the players don't have perfect information about the call. What they do have is a degree of belief about the call—sometimes they might see that a call is obviously wrong or right, whereas other times they could be entirely unsure.

This uncertainty is what makes challenging pitches tricky. If you lose a risky challenge early, you may forfeit the ability to make a sure-thing challenge later that alters the outcome of the game. It is not immediately obvious how to utilize the challenges as a resource to maximize win probability.

With that motivation, I develop a framework for testing challenge strategies. The metric for comparison will be the win probability (between two league-average teams) when each team uses its own strategy. To do so, I model the game as a Markov decision process (MDP).

2 Defining The MDP

The Markov process is a nice way of modeling the game because we can assign the value at the end states to be 1 if the home team wins and 0 if the away team wins. The value of all the intermediate states can then be interpreted as the win probability of the home team at that given state. Using the long-run win probability found here, I found that the values obtained from the (non-challenge) Markov chain I develop match closely with the historical data, so there is good reason to believe this is an accurate model of the game. [1]

Now, let's define the notation. Some of this is common for MDPs, and some I've included to fit the problem at hand.

Notation

s	Game state, defined by inning, half inning, balls, strikes, outs, bases, run differential, home challenges, and away challenges
s_0	Initial game state
s'_{no}	New game state after a called ball or strike and no challenge
s'_{success}	New game state after a called ball or strike and a successful challenge
s'_{fail}	New game state after a called ball or strike and a failed challenge
$V(s)$	Value of game state, s. Equivalent to win probability of home team
$V^{\pi}(s)$	Value of game state, s, under challenge strategy, π
x	Distance of pitch from the center of plate, as explained in Section 3
$\alpha(x)$	Confidence of player that pitch was called incorrect (given location of pitch x)
$\alpha^{*}(s;\pi)$	Least confidence at which a player will challenge under policy π at state s

[1] The win probability calculator can't be used for all states because some states are quite rare, so the sample size quickly becomes small. Also, the past games don't have challenges in place, so it would be impossible to tell the difference in win probability for two states that are exactly the same besides the number of remaining challenges.

With the game state defined, we can specify how the game evolves pitch by pitch. For each count, transition probabilities for the following possible outcomes are computed: ball, strike, single, double, triple, home run, out. As an example, consider the state

$$s = \{\text{5th inning, top, 2 balls, 2 strikes, 0 outs, runner on 1st, +2 runs, 2 challenges, 2 challenges}\}$$

The corresponding probabilities for every outcome are found given that the count is 2-2. There is a 40.6% chance a ball is thrown and the state transitions to the same as the previous, but with 3 balls. There is a 1.3% chance a home run is hit and the state transitions to

$$\{\text{5th inning, top, 0 balls, 0 strikes, 0 outs, no one on, +4 runs, 2 challenges, 2 challenges}\}$$

And so on for each outcome. There are some other interesting tests besides pitch challenging that could use this framework. For instance: what if two teams are equal, except team A strikes out slightly more? What if team A strikes out slightly more, but also hits more home runs? What if team A throws more strikes, but team B knows this, and swings more? Because baseball has such a well defined transition from state to state (whenever a pitch occurs), these sorts of tests lend themselves nicely to the MDP analysis.

3 Introducing Challenges

With challenges, additional events can occur after a called ball or strike: (1) no challenge, (2) a successful challenge, or (3) a failed challenge. To incorporate challenges into the model, the transition probabilities for these events must be computed.

Imagine you are a batter, pitcher, or catcher. The pitch is thrown, and you view it land a distance away from the center of the plate x. The umpire calls "ball" or "strike." You will have some confidence that the pitch was called incorrectly given the location of the pitch, $\alpha(x)$, which is the same as the subjective probability that a challenge will be successful.

You must then decide whether or not to challenge the call, i.e. if $\alpha(x) \geq \alpha^*(s;\pi)$. See that $\alpha^*(s;\pi)$ is only a function of the challenge policy, π, that the player is employing, and the current state, s. It is not a function of x, because it exists before the pitch is thrown. We can apply the inverse of $\alpha(x)$ to this $\alpha^*(s;\pi)$ to get x^*, the distance such that

$$\alpha(x^*) = \alpha^*(s;\pi).$$

Then the probability of a challenge can be written in terms of x.

For the batter,

$$\mathbb{P}(\text{challenge} \mid \text{called strike}) = \mathbb{P}\big(\alpha^*(s;\pi) < \alpha(x) \mid \text{called strike}\big) = \mathbb{P}(x^* < x \mid \text{called strike}).$$

For the pitcher/catcher,

$$\mathbb{P}(\text{challenge} \mid \text{called ball}) = \mathbb{P}\big(\alpha^*(s;\pi) < \alpha(x) \mid \text{called ball}\big) = \mathbb{P}(x^* > x \mid \text{called ball}).$$

The inequality flips because a batter will challenge so long as the pitch is far enough away from the heart of the plate, whereas the pitcher/catcher will challenge so long as the pitch is close enough to the heart of the plate.

The probability of a challenge succeeding for a given instance is just $\alpha(x)$. Thefore the probability of a challenge succeeding for any pitch, conditioning on the fact that a challenge was used, is

$$\mathbb{P}(\text{success} \mid \text{challenge}) = \mathbb{E}_x\big[\alpha(x) \mid \alpha^*(s;\pi) < \alpha(x)\big].$$

For the batter this becomes

$$\mathbb{E}_x[\alpha(x) \mid \text{called strike and } (x^* < x)],$$

and for the pitcher/catcher,

$$\mathbb{E}_x[\alpha(x) \mid \text{called ball and } (x^* > x)].$$

As we will see shortly, the fact that the expected value is conditional on how the ball was called by the umpire is very important. The distribution of a ball's location given it is called a strike, for instance, skews heavily towards the center of the plate, making it much less likely it was truly a ball. The umpire calling a pitch a certain way gives you some significant information about the true location of the pitch, which is unsurprising.

We can see now that the probabilities we are wanting to calculate are purely functions of x, the location of the pitch. Luckily, there is an abundance of pitch data available from the MLB.

4 Pitch Location and Confidence Distributions

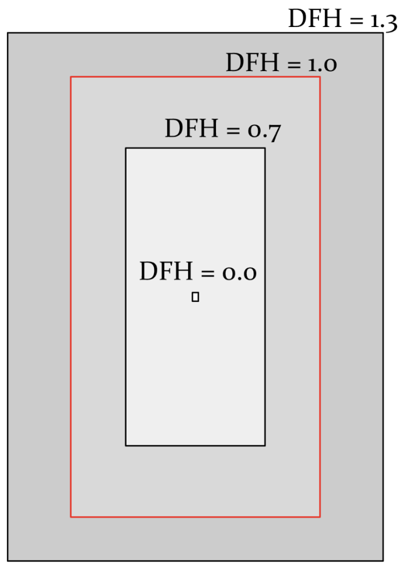


Figure 1: Definition of distance from heart (DFH) relative to the strike zone.

Up to this point I have used x to describe the distance of a pitch from the center of the plate, but I haven't described how to quantify it. I use the term *distance from heart* (DFH). To illustrate, see **??**. The red box is the strike zone for a particular batter.

If a pitch lands exactly on that red perimeter, its DFH is 1.0. Now consider a larger rectangle with width and height 30% bigger than the strike zone. If a pitch lands on that perimeter, its DFH is 1.3. At the center of the plate, DFH is 0.0. One way to interpret DFH is: if you draw a scaled strike zone centered at the plate such that the pitch lies on its boundary, DFH is the ratio of that scaled zone's dimensions to the true strike zone's dimensions.

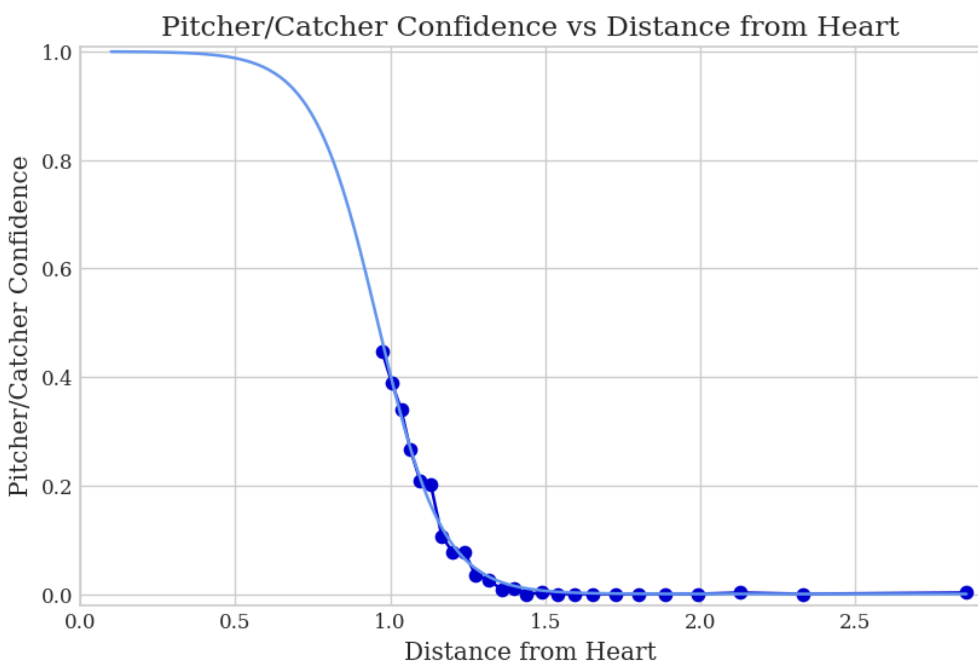
To see how this metric is useful, see Figure **??**, in which umpire error is plotted as a function of DFH.

As the pitches get closer to the border of the strike zone, the umpire error increases, until it reaches a maximum and decreases as the pitches get further away. In itself this plot is quite interesting—umpires are known to have an average accuracy around 94%. However, we see in the plot that this mostly comes from "easy to call" pitches, close to the center or far outside the plate. In fact, when pitches are borderline, the accuracy is only around 55% to 65%.

I use this as a starting place to find $\alpha(x)$, the players' confidence as a function of location. If a called strike appears right on the border of the strike zone (DFH of 1.0) a batter can be about 45% sure the umpire's call was incorrect. When a called strike appears even further from the plate the batter should be even more confident in a challenge, and if it appears closer to the inside the batter should be less confident. The reverse logic applies to pitchers/catchers and called balls. I fit a logistic curve to get these functions for both batter and pitcher, seen as the light blue lines in

Figure 2: Umpire error as a function DFH.

Figure **??** and Figure **??**.



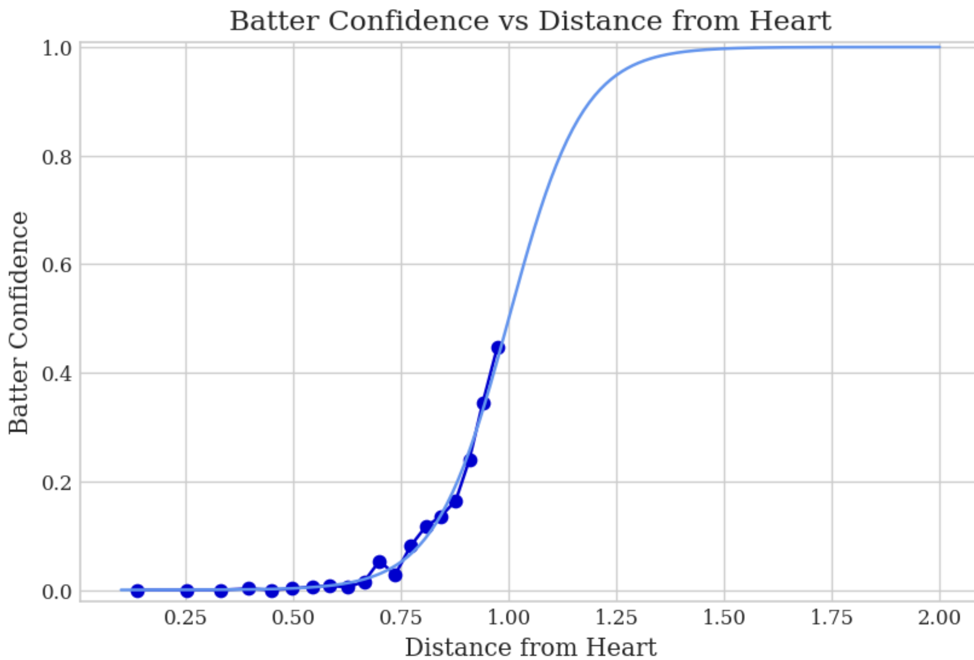


Figure 4: Batter confidence function.

the Markov model by deriving: (1) the probability of a challenge and (2) the probability of success conditional on a challenge. The derivations are also in the Appendix.

5 Defining Policies

Using the MDP, I now run tests to see how certain challenging policies compare to each other. We will use two teams are exactly equal in ability, only differing in policy. For example, one team might follow the policy "challenge whenever your confidence of being right is greater than 75%". There are more complicated policies one can come up with. If you were a computer playing baseball, you might want to challenge when the following is true (as the home team):

$$V^{\pi}(s_{\text{no challenge}}) < \alpha V^{\pi}(s_{\text{challenge win}}) + (1-\alpha)V^{\pi}(s_{\text{challenge loss}})$$

That is, given opposing challenge policy, π, your expected value of the new game state after challenging is greater than the value of the game state when you don't challenge. You can derive only when they are increasing the expected value of the state. Using that, you now have a $V(s)$ for every game state. The policy you would then define, independent of your opponen't policy, follows

$$\alpha^*(s) = \frac{V(s_{\text{no challenge}}) - V(s_{\text{challenge loss}})}{V(s_{\text{challenge win}}) - V(s_{\text{challenge loss}})}$$

Of course these two previous policies are impossible to implement—computers don't play baseball—but they can be helpful as a benchmark. In testing, I found that these the self-consistent policy is a very close approximation to the first (within hundredths of a percent). I will reference both strategies as the Computer Baseline (CB) and just report one value. For instance, Table **??** shows a simple confidence-only policy against the CB policy.

Table 1: Simple Confidence Policy Overview

Strategy	**Win % vs. CB**
Confidence over 40%	47.6
Confidence over 50%	48.3
Confidence over 60%	48.4
Confidence over 70%	47.3

Notice from this table that, if you are only considering confidence, a good lower bound is around 50–60%, as that alone will get you within two percent of CB. This is perhaps surprising, one might guess that a good rule of thumb would suggest being 80% or even 90% sure, so you don't waste your challenges. I think there are two reasons for the result. The most important of these is that, even though the lower threshold may be 50%, this does not mean every challenge has a 50% chance of succeeding, in fact it is significantly higher. You may challenge some pitches at 50% confidence, but you will also challenge some pitches at 75%, or close to 100%. The second factor is that, if you are waiting to challenge pitches with 90% confidence, you may never even get to use your challenges, as those pitches are quite rare. Umpire accuracy, as previously seen, is very good for pitches well-outside or well-inside the zone.

At this point we can also answer the question "How valuable is a challenge?" We can compare $V(s_0)$ values from the self-consistent policy to $V(s_0')$ and $V(s_0'')$, where s_0' indicates beginning the game with one challenge and s_0'' indicates beginning the game with no challenges. Of course $V(s_0) = 50\%$, as we have two equal teams playing and do not account for home field advantage. It is also found that $V(s_0'') = 43.3\%$ and $V(s_0') = 47.6\%$. For two otherwise evenly-matched teams, possessing challenges can *significantly* tip the odds.

When defining a challenge policy, it is tempting to include many rules and sub-rules so that the policy can accurately capture the leverage of a pitch. However, one can imagine that after more than a few rules the policy will be infeasible to implement. For that reason, I choose a handful of policies that use small sets of rules and confidence levels that are easy to interpret. The confidence levels are broken up into high, medium, and low:

$$\alpha_H = 0.8 \qquad \alpha_M = 0.5 \qquad \alpha_L = 0.25$$

In English, to challenge at α_H you need to be very sure a pitch was called incorrectly, at α_M you need at least a coin flip, and at α_L one should challenge if there's any chance at all the pitch was called incorrectly. I think these would be easy enough to implement accurately without having to think too hard about what makes an 80% chance or 25% chance. The selection of policies I have chosen to test are in the following table.

Table 2: Heuristic three-level challenge policies. Unless otherwise stated, the policy uses α_M.

Policy	Set $\alpha^*(s) = \alpha_H$ (conservative) when:	Set $\alpha^*(s) = \alpha_L$ (aggressive) when:
π_{inning}	inning ≤ 3	inning ≥ 8
π_{count}	balls ≤ 1 and strikes ≤ 1	balls ≥ 2 and strikes ≥ 2
π_{outs}	outs $= 0$	outs $= 2$
π_{runners}	bases empty	runner in scoring position (RISP)
$\pi_{\text{mix 1}}$	inning ≤ 3	bases loaded or inning ≥ 8 or full count
$\pi_{\text{mix 2}}$	inning ≤ 4	inning ≥ 8 or full count
$\pi_{\text{mix 3}}$	outs $= 2$ and bases empty	(inning ≥ 6 and strikes $= 2$) or full count
$\pi_{\text{mix 4}}$	balls ≤ 1 and strikes ≤ 1	(inning ≥ 5 and (strikes $= 2$ or balls $= 3$)) or full count
$\pi_{\text{mix 5}}$	balls ≤ 1 and strikes ≤ 1	balls $= 3$ or strikes $= 2$

6 Results and Recommendations

In the following table I report the home team's win probability as a percentage when the home team uses selected policy π and the away team uses baseline policy π_{base}. The baseline π_{high} is α_H for all states, and π_{CB} which is previously defined as CB.

Table 3: Performance of selected challenge policies against baseline opponents.

Heuristic policy π	Win% vs. π_{high}	Win% vs. π_{CB}
π_{inning}	52.1	48.1
π_{count}	52.6	48.6
π_{outs}	50.7	46.7
π_{runners}	51.8	47.8
$\pi_{\text{mix 1}}$	52.2	48.2
$\pi_{\text{mix 2}}$	52.1	48.0
$\pi_{\text{mix 3}}$	52.6	48.6
$\pi_{\text{mix 4}}$	52.7	48.7
$\pi_{\text{mix 5}}$	52.2	48.2

The three best performing policies are π_{count}, $\pi_{\text{mix 3}}$, and $\pi_{\text{mix 4}}$ with an expected win percentage of 52.6%, 52.6%, and 52.7%, respectively against the high confidence strategy. They also perform the best against the CB policy. In fact, you can see the performances against the CB policy are essentially that of the high confidence strategy shifted down 4.0%. Because the way teams use challenges affect the game (essentially) independently of each other, this result does make sense.

The π_{count} policy was used for one further test against a baseline of π_{med}, which is α_M for all states. The win probability for π_{count} was found to be 50.4%. Recall from earlier that this π_{med} policy also had an expected win percentage of 48.3% against CB, which is significantly better than almost all devised strategies.

Recommendations. The results suggest two simple and meaningful recommendations.

First, the default confidence at which a player should challenge a call is 50%. This alone will perform better than nearly all other simple strategies, and the effect could be a difference of 2%–2.5% in win probability, or 3–4 games over the course of a whole season. That is very notable, especially given the millions of dollars teams pay for wins in the player market.

The second recommendation is that if a team is looking for an even further, albeit modest (0.4%) edge using the challenge system optimally, look no further than the count. Simply adopt the count policy which follows the rules:

1) Challenge conservatively ($\geq$ 80% confidence) early in the count (0–0, 1–0, 0–1, 1–1).
2) Challenge aggressively ($\geq$ 25% confidence) late in the count (2–2, 3–2).
3) Challenge at $\geq$ 50% otherwise.

While you can try and scrape out more tenths of a percent in win probability with more complicated systems, those 3 easy rules will give you as much of an edge as you can practically hope for.

Finally, it is important to mention that another natural set of policies are ones that save challenges for certain moments. For instance, "no challenges before the 5th inning," or "no challenges with less than 2 outs," or "no challenges unless there are runners in scoring position." These are very bad policies. In comparison with π_{high}, they generally perform worse by 1% to 2%. This is potentially important because, without doing any analysis, these sorts of policies could seem like a nice idea. The good news is that using the recommended approaches against teams with "no challenge" policies will leverage an even larger advantage.

7 Appendix

Pitch Location Distribution Examples

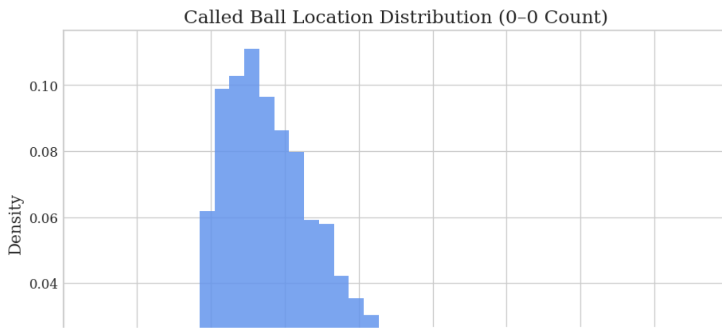


Figure 7: Called ball location distribution at 0–0 count.

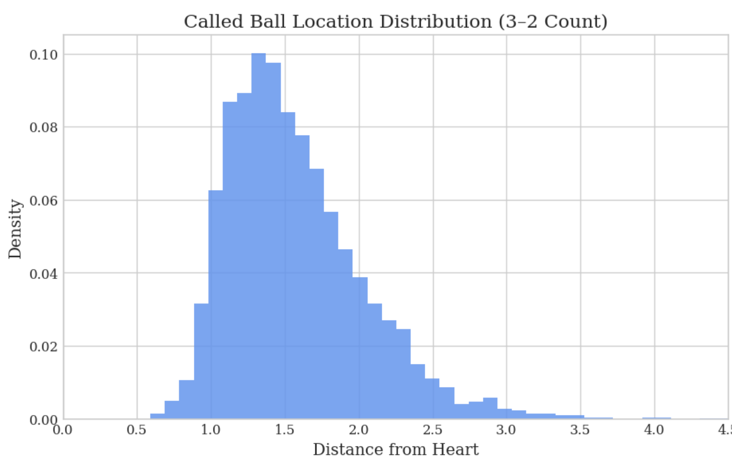


Figure 8: Called ball location distribution at 3–2 count.

Deriving $\mathbb{P}$(Challenge) and $\mathbb{P}$(Success)

Consider $\mathbb{P}(\text{challenge} \mid \text{called strike}) = \mathbb{P}(x^* < x \mid \text{called strike})$. First, we need x^* in terms of what we know. This requires the functions $\alpha_{\text{pitcher}}(x)$ and $\alpha_{\text{batter}}(x)$ seen as curves in Figures 3 and 4, respectively. We have for the batter

$$\alpha_{\text{b}}(x) = \frac{1}{1 + \exp\left[-k(x - x_0)\right]}$$

with $x_0 = 1$ by inspection and k being solved for with a curve-fitting function. For the pitcher we have

$$\alpha_{\text{p}}(x) = \frac{1}{1 + \exp\left[k(x - x_0)\right]}$$

with $x_0 = 0.96$ by inspection and k being solved for with a curve-fitting function.

We solve for x^* t468g51 Tf 1 0 0 9 Tf8s68g5Tf 76.371 -1.636 Td [7196)4.2 0 0 1 299.413 224.488 Tm [(c6tf 1 0 0 1 131.

that we have x^*, calculating $\mathrm{P}(x^* < x \mid \text{called strike})$ (or $\mathrm{P}(x^* > x \mid \text{called ball})$ for the pitcher) is just using the data of called strikes (balls) on a given count, and finding the frequency at which the pitch actually lands outside (inside) that x^*.

Now consider

$$\mathbb{E}_x[\alpha(x) \mid \text{called strike and } (x^* < x)] .$$

I'm going to leave this only in terms of the batter's perspective, but it should be clear how to apply the same for the pitcher/catcher. Let's say each data point of a called strike's location is x_i. To find the expected value we want, we can calculate

$$\sum_{i:x_i > x^*} p(X = x_i)\alpha_b(x_i)$$

Because we have one of each data point, this reduces to

$$\frac{1}{N} \sum_{i:x_i > x^*} \alpha_b(x_i)$$

Where N is the number of called strikes s.t. $x_i > x^*$. So the expected confidence we want reduces to taking the mean of the confidence values of all called strikes beyond x^*.