Visualizing Pitches from the 2024 Major League Baseball Regular Season

Ian Curtis, Student and Jonathan Leidig, Professor

Abstract— Sports analytics and big data go hand in hand, espeically in Major League Baseball. MLB teams, coaches, players, and third party companies and fans spend countless hours analyzing MLB data in an attempt to model and visualize performance. The app presented in this report focuses on MLB pitchers in an attempt to help MLB batters identify patterns in the pitchers they are about to face.

Index Terms—Baseball, MLB, sports analytics

1 Introduction

Tremendous effort has been put into the world of sports analytics, including data collection, storage, analysis, and interpretation to benefit teams, players, coaches, fans, stakeholders, and more. Major League Baseball (MLB) is no stranger to data analytics and is certainly a part of the "big data" industry: data related to nearly all aspects of the game is collected and used to make decisions and plan for future games [2, 4]. Unsurprisingly, MLB has their own data analysis team, as do each of the 30 individual teams (who are trying to gain an advantage over their opponents) [11]. Players do not have the time or expertise to sift through all the petabytes of data collected on their and their opponents' performance to pull out relevant insights or patterns. As a result, data scientists are needed to help players and managers visualize data and make predictions so that they can turn that information into an action item for future games. In the words of Thomas Miller: "Data do not speak for themselves. Useful predictions do not arise out of thin air. It is our job to learn from data and build models that work." [8]

Good models and useful predictions start and end with good visualizations. With the mass amounts of data stored, it is usually recommended to begin analyses by exploring the data visually in an attempt to identify initial patterns (which in turn can lead to more detailed analysis and modeling) and to present the data in an aggregated manner to interested parties. Visualizations are also helpful to show the results of models (such as prediction accuracies or clustering patterns) and other insights post-analysis.

This report focuses primarily on the former type of visualization. Using MLB Statcast data from the 2024 regular season, we build a web application using an R Shiny interface to help visualize pitches and pitch types. Using the zooming navigation method paired with the chainsaw overview method, the app allows users to apply a number of filters to view charts of a specific pitcher's pitches.

2 Dataset Details

The data used for this paper was manually compiled from MLB's online Statcast database [4, 9]. While this site certainly doesn't include all variables and information that MLB records from games, there are enough variables to analyze. It is important to note that Statcast also contains player visuals, one of which is a Pitcher Visualization Report [3, 7]. This is a great tool and meets many of the visualization strategies and recommendations provided in this class; however, it does not allow for variable selection and appears to be geared towards baseball fans rather than current MLB players. In fact, most of the literature in the baseball analytics field focuses on a specific piece of MLB data and requires an understanding of the complex mathematics underlying the research. These articles don't always include visualizations and are not geared towards a general audience [12, 13, 14].

For this report, we use all pitch-by-pitch data from major league pitchers in the 2024 regular season. To get the data, the following filters were used:

1. Player Type: Pitcher
2. Season: 2024
3. Season Type: Regular Season

The "Game Date >=" and "Game Date <=" filters were used to download one day's data at a time by selecting Search and then the File/Chain icon on the right of the results (which gives a CSV file). These files were then merged in R and variables not used in the app were filtered out. The final dataset used contains 709,510 observations (pitches) and 47 variables. The code used for initial data cleaning and preparation can be found on the author's GitHub page and the app itself is located online. The actual dataset used in the app is too large to share, but the pieces and code used to construct it are on GitHub as well.

3 Audience & User Persona

Although multiple user groups may be interested in this application and/or the information therein (such as baseball fans or the media), the app was designed with MLB batters in mind. The goal is to help provide useful and insightful information about a major league pitcher quickly to allow batters to identify trends to better prepare to face a certain pitcher in a future game or in a certain scenario.

More specifically, the app is geared towards batters who are new to the major league scene. It's likely that these players have been deeply involved in the sport for many years, playing through high school, college, and the minor league baseball (MiLB) system. There certainly is no lack of talent in each of these different "sectors" of baseball; however, the major leagues take baseball to the next level. The best of the best play in the MLB, from rookies to veterans, and it can be daunting for a new player to join a major league team and face a professional pitcher. Moreover, these batters do not want to disappoint their teams or make their managers regret calling them up to the major leagues. They (like most batters) wish to avoid hitting slumps and excessive strikeouts and don't want to be embarrassed by pitchers who catch them off guard. The results of this paper aim to assist these new MLB batters in learning a little more about the pitchers they are about to face before they play in a live game. We also recognize that veteran batters may be interested in learning more about new pitchers to the major league scene.

This app does not attempt to incorporate any predictive modeling or other advanced statistical methods such as hypothesis testing. These results go against the main purpose of the app: to provide useful information quickly. Nevertheless, there are certain questions that this app attempts to provide an answer to, through dataset filtering and visualization:

1. Does the release point (position in space) of the pitch vary by type of pitch / by pitcher?
2. What are the locations where each pitch type crosses the plate or is caught by the catcher?

3. What kinds of pitches does a pitcher throw? Can we informally predict which pitch will be thrown at any given time?
4. How fast does the pitcher throw? Is speed related to any other information about the game (such as inning)? What kinds of pitches result in hits? In outs?
5. Where do a pitcher's statistics fall in relation to the other pitchers in the league?

4 Application Interface and Charts

The application consists of five major portions: a sidebar, a general information panel, and three reactive, interactive plots (Fig. 1). The sidebar contains 13 different filtering options, 1 option to add color to the plots, 1 option to change the third plot type, an update button (which triggers plots to generate), and a reset button (which sets all filtering options back to their defaults). The dataset used for this project is considerably too large to explore easily; so, to help support the project's goal of easy-to-access information, MLB batters may use the filters to zoom in to see specific data on a by-pitcher basis. More

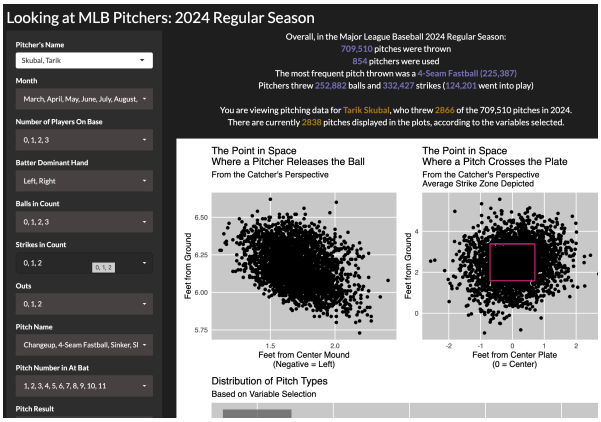

Fig. 1, The app interface (for pitcher Tarik Skubal)

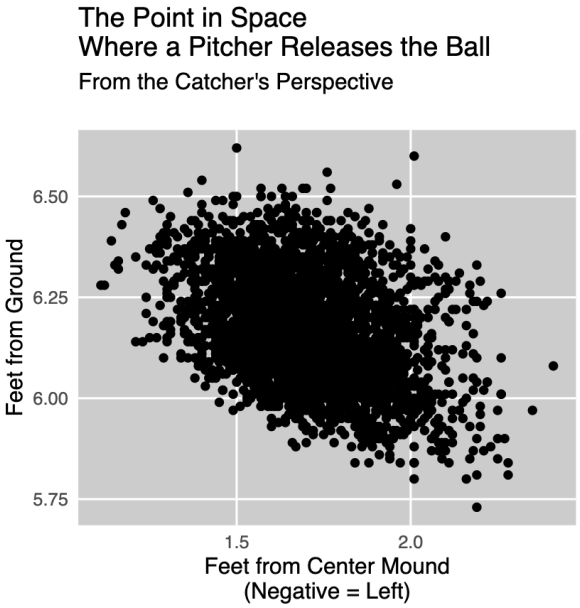

Fig. 2. Tarik Skubal's release point diagram

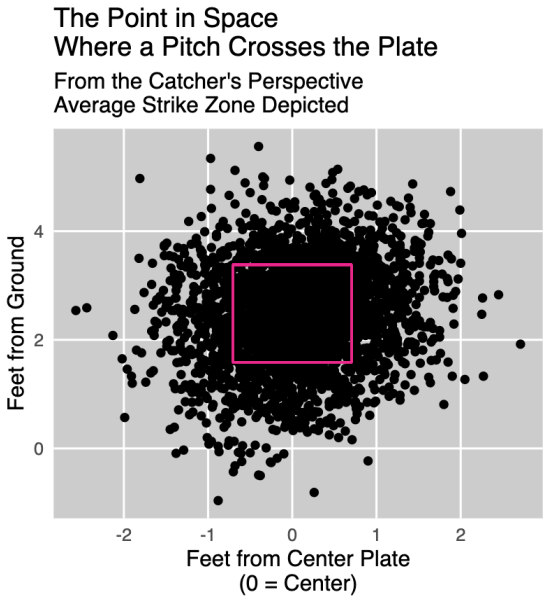

Fig. 3. The point at which Tarik Skubal's pitches cross the plate.

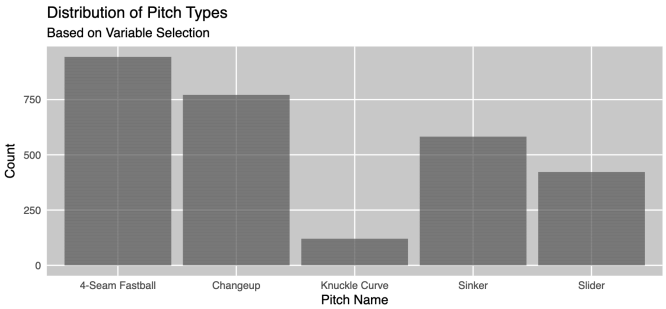

Fig. 4. Tarik Skubal's pitch types

filters are provided to help support zooming in further, if necessary. For instance, some batters may be interested in how a pitcher performs under pressure (2+ players on base, 2 outs in the inning, pitcher's team currently winning) or what a pitcher may tend to do in the first pitch in an at bat (pitch number in at bat = 1). To aid in insight discovery, users can also select a variable that they would like the plots to be colored by.

Once the Update Plots! button is selected, the app filters the dataset to the selected variables and generates plots from that data. Fig. 1 shows a zoomed-out screenshot from the app after the Update Plots! button was selected for pitcher Tarik Skubal. At the top of the newly generated output is text placing the results in the context of the dataset (Research Question 5). Users will always see the following static information about the entire dataset:

- How many total pitches were thrown
- How many unique pitchers there were
- The most frequent pitch thrown
- The numbers of balls, strikes, and in play pitches

Then, users will see the following dynamic information based on the pitcher variables selected:

- The pitcher's name
- The total pitches the selected pitcher threw
- The total pitches currently represented in the plots

Fig. 2, Fig. 3, and Fig. 4 show zoomed-in versions of the three plots given from the app. From here, users can examine the plots and make adjustments to the filters, iteratively changing the variables and zoom level into the dataset. With all of the provided charts, the goal was not to persuade the user into believing anything; we simply wish to provide a way to visualize information about pitches and let the user produce the main takeaways.

Although this interface takes mostly a zooming approach to navigation, the added context details aim to give some context to the results bringing the interface close to a Focus + Context navigation. We relied on the chainsaw overview method to produce the charts as the data depicted was chopped from the overall dataset. Based on user selection, attributes are stripped or added, and the dataset is filtered to narrow down the number of data points displayed. All charts lean more towards the Edward Tufte proposition that plots should be simple and easy to understand with no deception (although some of the plots have an Tufte lie factor, see information below).

4.1 Chart 1: Pitch Release Point

Fig. 2 shows the first chart in the interface. This plot details the position in space where the ball left the pitcher's hand, as viewed from the catcher's perspective [9]. This means that release points to the *right* of 0 (the center of the pitching mound) are likely from *left*-handed pitchers and conversely release points to the *left* of 0 are likely from *right*-handed pitchers. There is a lie factor in this chart: the axes are not centered at 0 and for most pitchers will not even show x = 0 [6]. It turns out that most pitchers are remarkably consistent in release pitches around the same point in space. If the chart was left with an x-axis ranging from, say, -4 to 4 (feet from center mound) and a y-axis from 0 to 7 (feet from the ground), we would be left with a tiny cloud of points and quite a bit of extra, unused white space on the plot (Fig. 5, from the development version of the app). As a result, it was determined that increasing the lie ratio of the chart from the axes would allow users to get more insight out of the app as the chart would be zoomed in enough to see individual pitches.

A scatterplot was chosen for this chart as position is the first priority in Cleveland's rules for numeric data. We plot the coordinates of the release point of a pitch as a catcher would have seen the ball leave the pitcher's hand to allow for easy interpretation of the data.

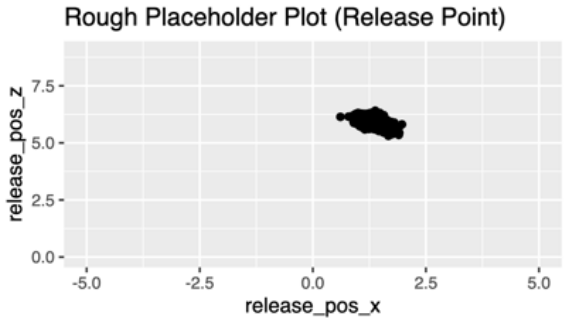

Fig. 5. A development version of the release point plot. Notice the small clump of points and large amount of the plot not being used.

4.2 Chart 2: Crossing Home Plate

Fig. 3 shows the position in 2-D space where the pitched ball crosses home plate, from the catcher's perspective. It is important to note that home plate is a five-sided polygon and the corresponding strike zone a pentagonal prism [10]. As a result, this chart is not completely accurate; however, it provides enough information to allow MLB batters to obtain some useful information. In this chart, x = 0 is the center of home plate and the y-axis shows feet from the ground.

Relying on the Gestalt Principle of Enclosure, this chart also shows the average strike zone for this pitcher, helping group together the data points that are part of the strike zone. It is important to note that this box was calculated for visualization purposes and not for exact delineation of balls vs. strikes. Official MLB rules state that if any part of a baseball crosses the plate and is inside of the strike zone, that pitch is a strike. The width of home plate is always 17 inches (1.41667 feet); this width is fixed in the chart displayed by the app [5]. The height of the strike zone is defined as the "midpoint between a batter's shoulders and the top of the uniform pants -- when the batter is in his stance and prepared to swing at a pitched ball". As such, the height of the strike zone will change slightly between batters. For each observation in the dataset, there is an associated value for the top and bottom of the strike zone. Based on the variables selected, the app calculates the average of all of these values and uses them to plot the top and bottom of the strike zone, respectively. As a result of this modified strike zone and the fact that we cannot accurately display the three-dimensional strike zone prism, we can consider this plot containing a lie factor of slightly greater than one (effect size in graphic greater than effect size in the data) [6].

4.3 Chart 3: Pitch Type Frequencies

Fig. 4 plots the frequencies of pitch types thrown by the selected pitcher. This is essentially a simple bar chart that adjusts as the dataset is filtered down. When a second variable is added, the chart follows Cleveland's Rules: the bars are colored based on levels of a chosen categorical variable (see below). Users also have the option to select Pitch Type Clusters as the third chart which plots each pitch's release speed and spin rate, forming clusters of pitch types. This is useful if the user is filtering the data set down to only a few pitch types and the bar chart becomes no longer informative.

4.4 Additional Chart Features

A major piece of these plots that has not been mentioned is the interactivity. All three of these charts are linked together: when the user hovers over any point in the top two charts or a piece of the bottom bar chart, the corresponding points/bar chunks are automatically highlighted on the other two plots. The user may also click on a specific point to keep that point highlighted, even when the cursor is removed which turns that point into a focal point (a distinct object). In addition to selecting individual points, users can hover over specific points on charts 1 and 2 to pull up a tooltip menu giving details about that specific pitch to help place that pitch in the context of the rest of that player's pitches. The tooltip statistics consist of pitch speed, pitch type, pitch result, inning, and month.

If the user chooses to select an extra variable to add (using the "Color Plots By…" selector), the charts will become colored by that variable, using the ColorBrewer colorblind friendly palette Dark2. All charts retain the same structure and convey the same information in addition to the added color (and corresponding legend). We use color to help users better understand which points are grouped together as the Gestalt Principle of Similarity indicates that items with similar characteristics are grouped together when perceived.

5 Case Study

Recall the original research questions from above:

1. Does the release point (position in space) of the pitch vary by type of pitch / by pitcher?
2. What are the locations where each pitch type crosses the plate or is caught by the catcher?
3. What kinds of pitches does a pitcher throw? Can we informally predict which pitch will be thrown at any given time?
4. How fast does the pitcher throw? Is speed related to any other information about the game (such as inning)? What kinds of pitches result in hits? In outs?
5. Where do a pitcher's statistics fall in relation to the other pitchers in the league?

To demonstrate the functionality of the web application and to provide an example set of answers to the above questions, we will now conduct a case study.

5.1 Case Study: Who Are We?

In this particular case study, we will approach the app from the point of view from a baseball player who has just been called up from the minor leagues. Let's call him Mike. His first game is tomorrow, and he is quite nervous. Not only that, but he will be facing Paul Skenes. Skenes is also a rookie but has already established himself as a dominant pitcher in the league. In fact, most batters in the league struggle against Skenes. Nevertheless, Mike wants to be as prepared as possible for this upcoming game. With so little time to research Skenes, he decides to use the app presented in this paper.

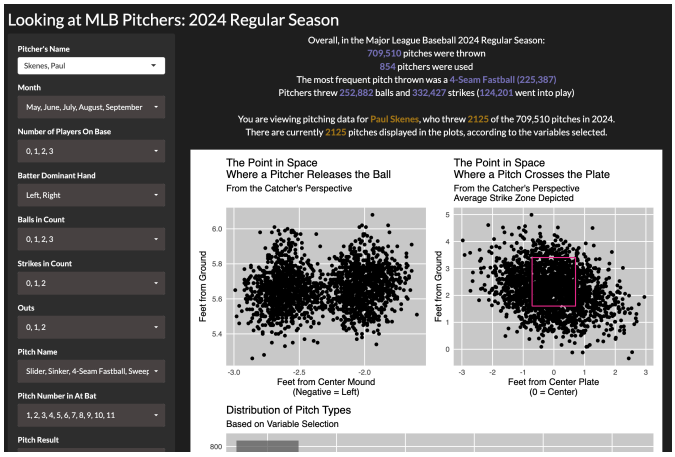

Fig. 6. Case study: Default application output for Paul Skenes

5.2 Case Study: Finding Insights

Mike starts by logging into the app. He isn't sure exactly what variables he wants to filter by yet so he first will run the app as is. He selects "Paul Skenes" from the player drop down menu and chooses Update Plots. Fig. 6 shows what Mike sees. It looks like Skenes threw 2125 pitches in 2024, a small fraction (~0.2%) of the over 709,000 pitches thrown across all 854 pitchers. Scrolling down, Mike sees that, like the majority of pitchers in 2024, Skenes threw more 4-Seam Fastballs than any other pitch type (Research Question 5). This is

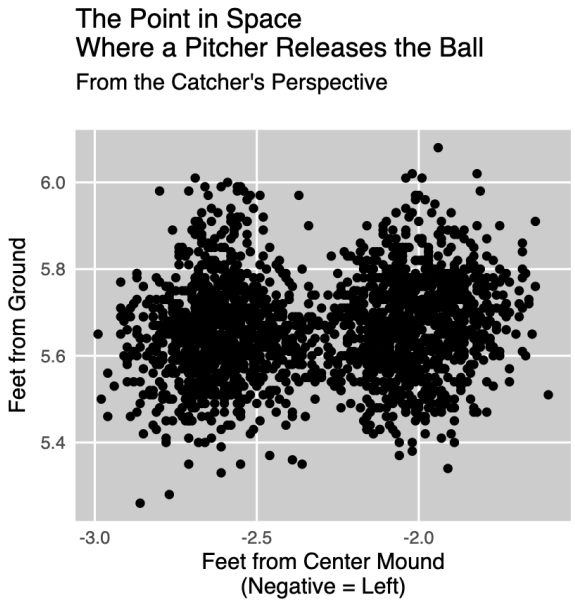

Fig. 7. Paul Skenes's release point diagram.

followed by a Sinker as second most frequent. He stores this in the back of his head: he would expect to see more fastballs and sinkers and should be prepared to get fooled by a sinker that looks like a fastball at first (Research Question 3). Mike's other big takeaway is that there appears to be two separate blobs of pitches in terms of when the ball leaves Skenes's hand (Fig. 7).

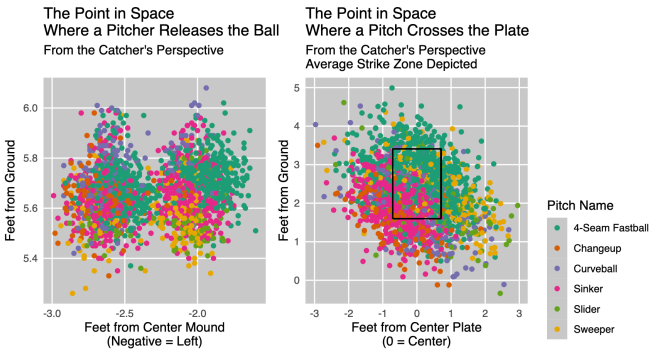

Fig. 8. Paul Skenes's pitches

This is quite interesting as most pitchers tend to be fairly consistent when releasing a pitch to avoid giving any indication that a certain pitch type is coming (see Skubal's plot from above). Seeing this, Mike wants to see if he can identify a reason for these two different release points. Also, as he sees the app and the possible variables to filter by, he becomes interested in Skenes's pitch patterns for the first pitch in an at bat and the pitches that took place in a high-pressure situation.

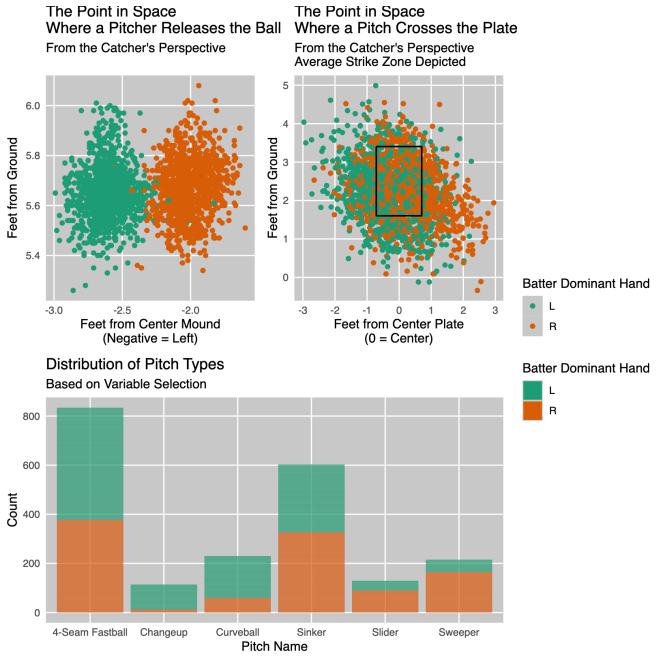

Fig. 9. Pitch distribution for Paul Skenes, colored by batter dominant hand.

Mike thinks that maybe the type of pitch affects the release point of the pitch. He uses the Pitch Name variable to select one type of pitch at a time. He sees that only changeups have a tendency to be release farther out to Mike's left, but all other pitches show the two blobs as before. Next, he looks at runners on base. Maybe Skenes has different release points based on how many runners are on base so that the baserunner(s) can't guess what pitch he's throwing. This time, instead of using the Number of Players on Base filter, he decides to use the Color Plots by… option to color by players on base (Fig. 8).

Still not much to see. All values for Players on Base appear in both of the blobs. He isn't sure what is leading to the two differing release points, so Mike changes the Color Plots by… option to each of the variables in the dropdown to see what happens. Finally, he finds the answer. When he colors the plots by Batter Dominant Hand, he sees a clear difference between the two sets of points (Fig. 9). It looks like Skenes has a slightly different release point depending on if the batter is left- or right-handed (Research Question 1). Mike is left-handed so, to investigate further, he uses the sidebar to filter to left-handed batters only and colors the plots by pitch name. Mike sees that he should expect pitches to be released about 2.5-3 feet left from center mound (Fig. 10). He also notices (by hovering over points) that if a pitch is released further to his right than expected, it is likely to be a 4-Seam Fastball that results in a ball or a foul. He takes note of this; if he can catch on to this information, he might be able to resist swinging to get a ball. Mike moves on to his other areas of interest.

How does Skenes perform in the first pitch of an at bat? Mike filters by left-handed batters and sets Pitch Number in At Bat to 1, coloring by Pitch Name. He sees that he is highly likely to receive a Fastball, Curveball, or Sinker as the first pitch (Research Question 3). He now filters by just these three pitches. Fig. 11 shows these results. Mike looks at the plot depicting the point at which the pitches cross the plate. He notices that the average strike zone box is not in the

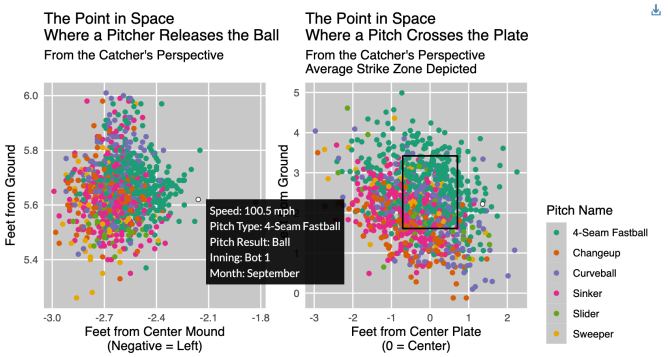

Fig. 10. Paul Skenes's pitches (to left-handed batters, colored by pitch name).

center of the plot–it is shifted up and to the right. This indicates that Skenes has a tendency to throw his first pitches either in the strike zone or more down and away from the batter (perhaps this is why Skenes changes his release point…he doesn't want to hit a batter). Mike also sees how first pitch Sinkers are usually down and away while Fastballs are more center or up and to the right. He notes that these Fastballs are also released towards the right, so if a ball seems to be coming right at him, it is likely a fastball and he might want to consider swinging (Research Question 2). To verify this, he quickly sets the Pitch Result filter to "Hit Into Play". Only 22 of Skenes's first pitches in an at bat to left-handed batters were hit into play and most of them were actually sinkers. He filters by Pitch Result set to "Swinging Strike" and sees that there were a handful of swinging strikes on these sinkers down and away. Mike thinks that maybe it's best to not swing at the first pitch and not take the risk of a sinker.

As a final example, Mike now wants to see how Skenes performs under high pressure situations. He decides that a "high pressure situation" means that there are 2 outs, more than 1 runner on base, and the game is tied. To get the corresponding plots for these parameters, Mike selects the "Reset to Defaults" button and filters for left-handed batters, 2 outs, 2 or 3 runners on base, and pitcher game state set to "Tied". Perhaps not surprisingly, Skenes does not have many pitches in this scenario (only 20). This indicates that he usually pitches well enough to not make it into high pressure scenarios (indeed only 27 pitches were thrown to right-handed batters in the same situation). There isn't a lot to go off of here, but Mike sees that in such a situation he would likely receive a fastball. He colors the plots by pitch name and sees that most of the fastballs land in the strike zone. However, hovering over the points, he sees that not a single fastball is pitched slower than 98 miles per hour (Research Question 4). Skenes throws *fast* in this scenario and, coloring the plots by pitch result, he usually gets out of the sticky situation: only two of the 20 pitches were hit into play (digging into the raw data, these two pitches both resulted in a field out). Mike understands that if he finds himself in this scenario, he should prepare for a fastball but that he shouldn't feel bad if he isn't able to convert a pitch into a hit.

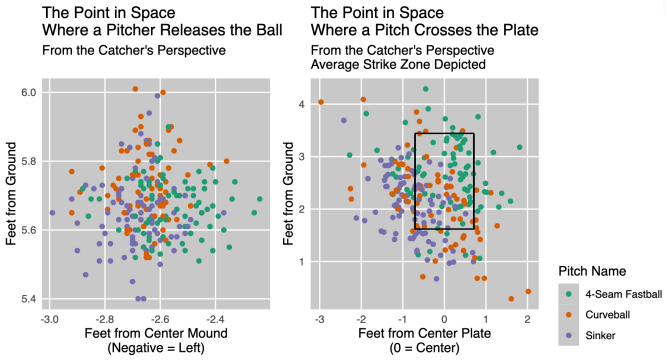

Fig. 11. Paul Skenes's first pitches in an at bat to left-handed batters.

6 Conclusion

Statistics and visualization play a large role in Major League Baseball. Individual teams and players are always looking for ways to out-play another team or player and rely on the mass amounts of data collected by MLB to model performance and make predictions. The app presented in this paper focuses on providing easy access to data through visualization, aimed at MLB batters, specifically those who are new to the MLB scene. It assumes a basic understanding of the type of data collected by MLB but does not rely on complex modeling, providing the freedom to the audience to select the variables they are interested in looking at.

References

[1] R. Kabacoff, *Chapter 13 Interactive Graphs | Modern Data Visualization with R*. Accessed: Nov. 18, 2024. [Online]. Available: https://rkabacoff.github.io/datavis/Interactive.html

[2] J. Mizels, B. Erickson, and P. Chalmers, "Current State of Data and Analytics Research in Baseball," *Curr Rev Musculoskelet Med*, vol. 15, no. 4, pp. 283–290, Apr. 2022, doi: 10.1007/s12178-022-09763-6.

[3] "Baseball Savant Visuals," baseballsavant.com. Accessed: Nov. 30, 2024. [Online]. Available: https://baseballsavant.mlb.com/visuals

[4] "Baseball Savant: Statcast, Trending MLB Players and Visualizations," baseballsavant.com. Accessed: Oct. 09, 2024. [Online]. Available: https://baseballsavant.mlb.com/

[5] "Field Dimensions | Glossary," MLB.com. Accessed: Nov. 30, 2024. [Online]. Available: https://www.mlb.com/glossary/rules/field-dimensions

[6] "Lie Factor - InfoVis:Wiki." Accessed: Nov. 30, 2024. [Online]. Available: https://infovis-wiki.net/wiki/Lie_Factor

[7] "Pitcher Visualization Report," baseballsavant.com. Accessed: Nov. 30, 2024. [Online]. Available: https://baseballsavant.mlb.com/player-scroll

[8] *Preface*. Accessed: Nov. 18, 2024. [Online]. Available: https://learning.oreilly.com/library/view/sports-analytics-and/9780133887402/pref01.html

[9] "Statcast Search CSV Documentation," baseballsavant.com. Accessed: Nov. 18, 2024. [Online]. Available: https://baseballsavant.mlb.com/csv-docs

[10] "Strike Zone | Glossary," MLB.com. Accessed: Nov. 30, 2024. [Online]. Available: https://www.mlb.com/glossary/rules/strike-zone

[11] A. Chen, "THE METRICS SYSTEM.," *Sports Illustrated*, vol. 125, no. 5, pp. 44–48, 20160822.

[12] K. Burris and J. Coleman, “Out of gas: quantifying fatigue in MLB relievers.,” *Journal of Quantitative Analysis in Sports*, vol. 14, no. 2, pp. 57–64, Jun. 2018.

[13] L. Ornelas, G. San Román, and I. Soria, “Most Effective Pitches Against Ronald Acuña Jr.,” *IEOM South American Conference Proceedings*, pp. 239–247, May 2024, doi: 10.46254/SA05.20240047.

[14] B.-J. Wen, C.-R. Chang, C.-W. Lan, and Y.-C. Zheng, “Magnus-Forces Analysis of Pitched-Baseball Trajectories Using YOLOv3-Tiny Deep Learning Algorithm.,” *Applied Sciences (2076-3417)*, vol. 12, no. 11, pp. 5540–5540, Jun. 2022, doi: 10.3390/app12115540.