Modelling tjStuff+ v3.0

Thomas Nestico 12 min read · Oct 27, 2024

You can find "Modelling tjStuff+ v1.0" here

You can find "Modelling tjStuff+ v2.0" here

All code for this project is available on GitHub

Introduction

At the start of 2024, I undertook the very interesting task of pitch modelling. More specifically, training a machine learning model which takes the physical characteristics of a pitch and then predicts the expected run value. This concept was not a novel one, as pitch models like this have grown in popularity in recent years. The most common vernacular used to describe such models is "Stuff", a relatively simple term to describe the effectiveness of a pitcher's arsenal.

I have learned a lot more about pitch modelling and machine learning since then, and decided that in April an update on my "tjStuff+" model was due.

Now fast-forward to the end of the 2024 MLB season, I planned to update my model once again. This time, I wanted to specifically walk through my code and my methodology when training and validating my model.

Let's begin!

Data Selection

My data selection will follow the same thought process as my earlier versions of the model. I will be training on 2020–22 Data, and then validating the model using 2023 data. Since I want to use tjStuff+ as a predictive model, I will validate the 2023 data using 2024 results.

Following the validation of the model, I will inject the 2023 data and then train the model again to determine the descriptive power of the model.

2024 data will not be added into the model until the commencement of the 2025 season.

All pitch data used for this project can be found here.

Data Preparation

The same data preparation from v1.0 was undertaken in v3.0.

> The physical characteristics of a pitch are well-defined and accurately measured, however these measurements are not normalized between pitchers of different handedness. This means that metrics such as Horizontal Release Point and Horizontal Break for left-handed pitchers would be scaled by a factor of -1 compared to right-handed pitchers. We can normalize these "mirrored" metrics so that during training, pitches thrown

> from either hand are on the same scale, which should improve performance.

Feature Selection

Funnily enough, my feature selection has mostly returned to v1.0. The biggest change from v1.0 and v2.0 was relating each pitch to a pitcher’s primary pitch instead of their primary fastball. In v3.0, each pitch will once again be compared to the pitcher’s primary fastball.

I decided to return to comparing against primary fastball because it returned better results during testing. I also made this change because using their primary pitch resulted in drastic changes in a pitcher’s grades depending on their primary pitch from start to start. I wanted to minimize this variation by restricting the primary pitch to fastballs.

Another change that I made was change Induced Vertical Break and Horizontal Break to Z Acceleration and X Acceleration respectively. This decision was made thanks to some insight from Max Bay.

The features (Definitions) for v3.0 are as follows:

start_speed

- The speed of a pitch as it is released from the pitcher’s hand, measured in miles per hour

spin_rate

- The rotation per minute of a pitch as it travels through the air

extension

- The release extension of a pitch measured in feet

ax

- The acceleration of the pitch, in feet per second per second, in x-dimension, determined at y=50 feet.

az

- The acceleration of the pitch, in feet per second per second, in z-dimension, determined at y=50 feet.

x0

- Horizontal Release Position of the ball measured in feet from the catcher’s perspective

z0

- Vertical Release Position of the ball, measured in feet from the catcher’s perspective.

spin_axis

- The Spin Axis in the 2D X-Z plane in degrees from 0 to 360, such that 180 represents a pure backspin fastball and 0 degrees represents a pure topspin (12–6) curveball

speed_diff

- For any given pitcher, the difference between release speed and their most used primary fastball average release speed.

az_diff

- For any given pitcher, the difference between az and their most used primary fastball average az.

ax_diff

- For any given pitcher, the difference between ax and their most used primary fastball average ax.

From my article on tjStuff+ v1.0:

> *My reasoning for including metrics related to a pitcher's fastball stems from the importance of sequencing in pitching. How different one's fastball is from their off-speed and braking pitches allows the pitcher to approach certain scenarios differently, and adds a layer of deception which makes it difficult for the batter to adapt.*

Target Selection

Run Value (RV) is the target in this model. Please refer to my article about modelling batter decision values for my methodology on preparing the Run Values for training. Code for generating the RV is includein my GitHub Repository for this project.

Model Selection

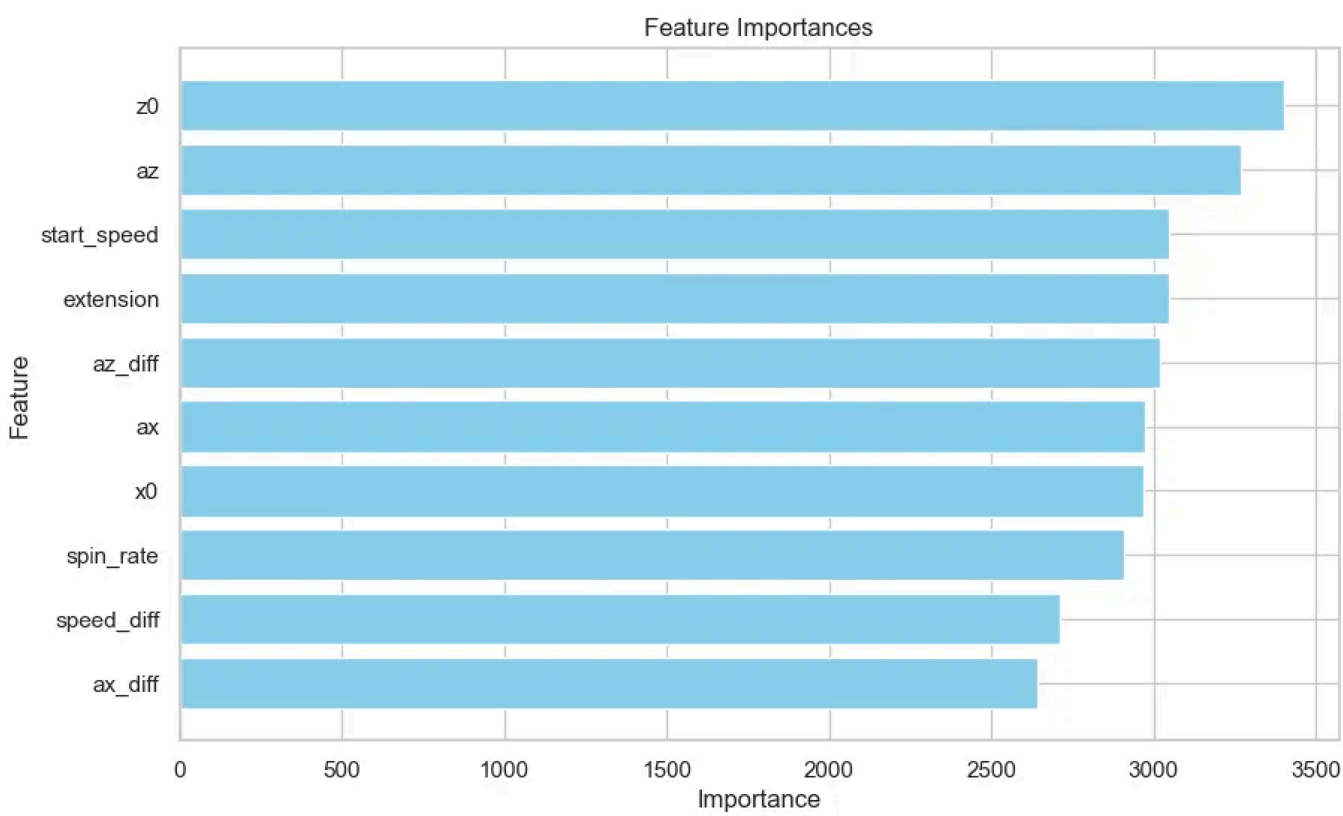
Now this is the biggest change between v2.0 and v3.0!

v3.0 uses a LightGBM Regressor, while v2.0 and earlier uses an XGBoosted Regressor. Both are popular gradient boosting frameworks, however the efficiency of LightGBM made it preferable when dealing with the large dataset of over 1.6 million pitches. Also through testing, LightGBM performed more favourably than XGBoost.

I also decided to apply Robust Scaler to the data, as it helped with outlier robustness and applied consistent scaling to the data. This helped with limiting the impact of outlier extension found in earlier versions of tjStuff+, and improved model performance.

Feature Importance

LGBMRegressor returns feature importance which helps us understand which features are most influential in making predictions. From our trained model, we see that Pitch Velocity and Z Accerlation are the some of the most impactful features, which is intuitive.



Feature Importance

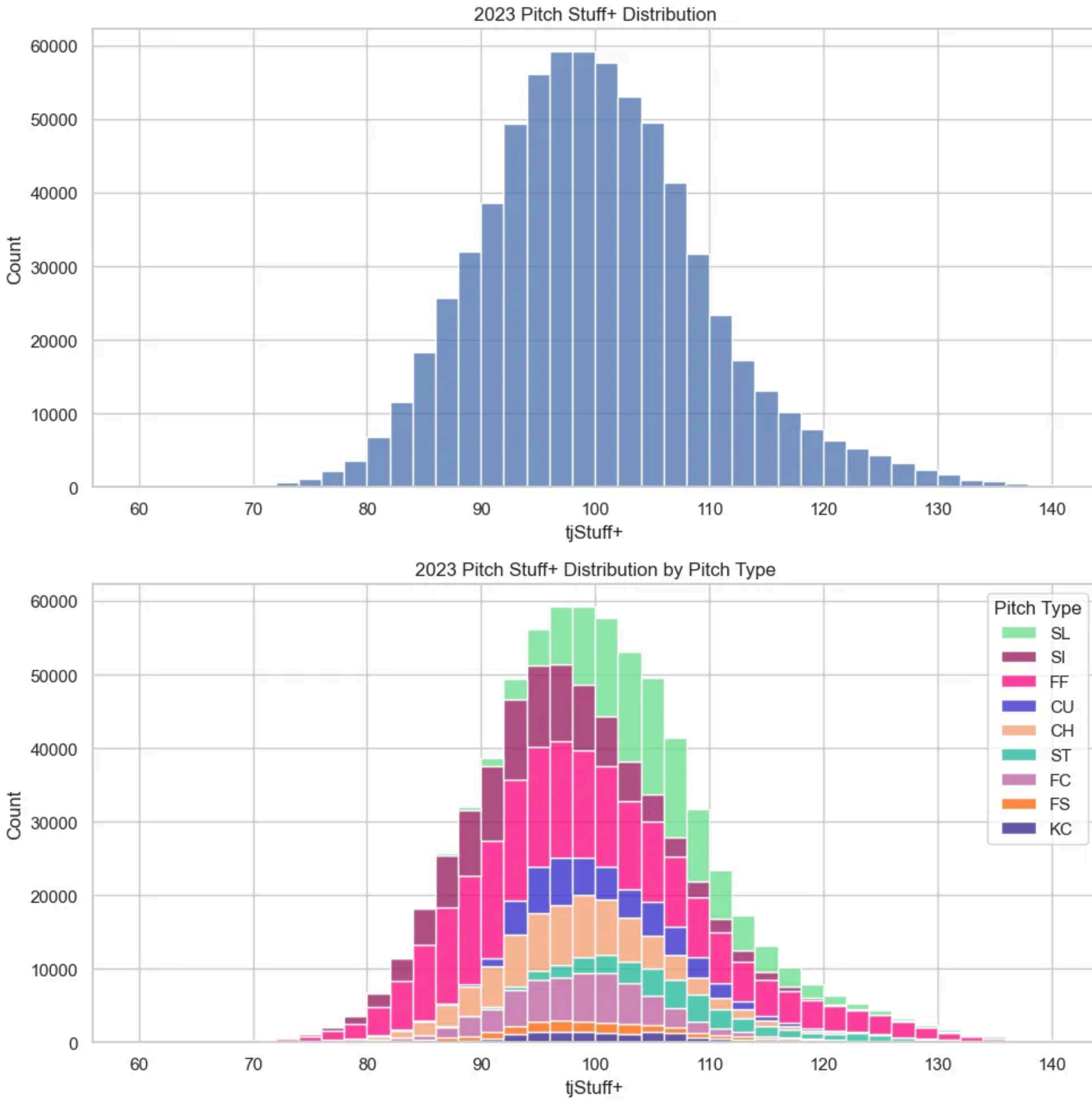
Validation

Calculating tjStuff+

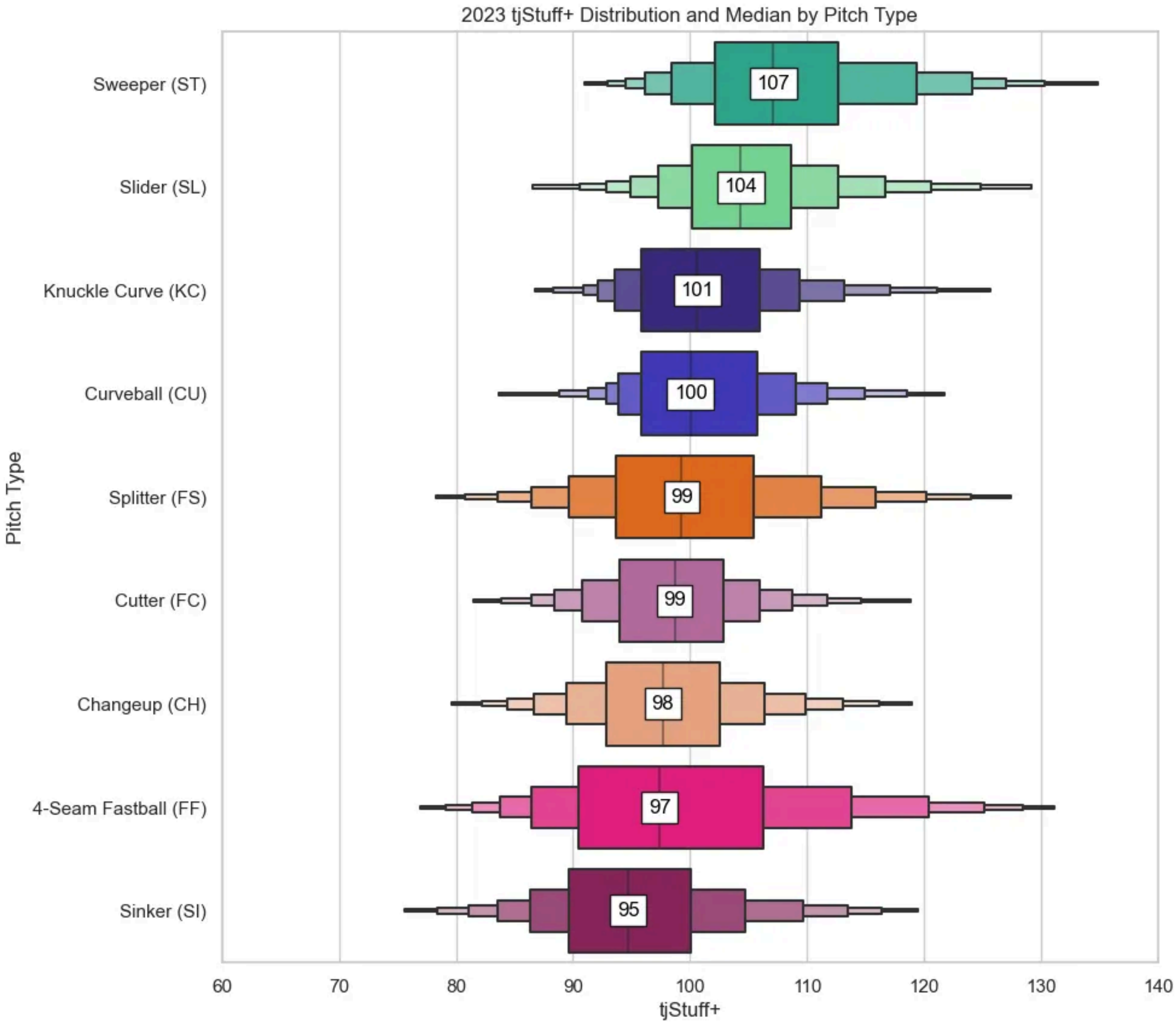
tjStuff+ v3.0 is calculated the same as v1.0:

> The output of my model is expected run value, which means that for any given pitch, the model can predict how effective that pitch is at limiting runs based on its physical characteristics. We can use a standardization technique to assist in comparing pitchers and pitches to one another. This is where the calculation of tjStuff+ arises.
>
> tjStuff+ is similar to the prospect tool grade scale. The prospect tool grade is a normal distribution which uses 50 as the average and 10 as the standard deviation. This means that a prospect with a “60 Grade” hit tool, has a hit tool 1 standard deviation above the mean, which would slot them approximately into the 84th percentile. Increase that to a “70 Grade” hit tool, and now the prospect sits at the 97th percentile of hit tools. tjStuff+ follows this same standardization, but uses 100 as the mean and 10 as the standard deviation

The following plots illustrate the distribution of single-pitch tjStuff+ for the 2023 season.



2023 Pitch Level tjStuff+ Distribution



2023 Pitch Type tjStuff+ Boxen Plots

Comparing 2023 vs 2024

To validate the model, we will predict tjStuff+ values on 2023 data and then calculating the correlation of tjStuff+ to 2024 results. The results we will use are FIP, wOBA, and K-BB%. We will also calculate tjStuff+ on 2024 data to evaluate the 'stickiness' of the metric.

Thanks to the Fangraphs API, we can easily grab 2024 MLB Pitcher Results. I also downloaded Pitching wOBA data from Baseball Savant. With the 2023 and 2024 Data loaded into a DataFrame, we can calculate correlations to test predictiveness and stickiness.

Predictiveness

Using a sample size of 100 pitches, tjStuff+ performs well compared to conventional metrics. wOBA is an extremely important metric to consider when predicting future performance, and tjStuff+ is the most effective in a sample of just 100 pitches.

```
Minimum Pitches: 100
Sample Size: 540
Average Pitches: 1096
Correlation between 2023 and 2024:
```

	2024 FIP	2024 wOBA	2024 K-BB%	2024 tjStuff+
2023 tjStuff+	0.26	0.27	0.31	0.85
2023 FIP	0.33	0.25	0.18	0.08
2023 wOBA	0.22	0.21	0.24	0.16
2023 K-BB%	0.27	0.24	0.26	0.16

2023 vs 2024 Correlation Matrix

Stickiness

With a correlation of 0.85 between 2023 tjStuff+ and 2024 tjStuff+, it is reasonable to say that tjStuff+ is a “sticky” statistic. The stickiness of tjStuff+ is desirable as it means that a player likely to attain a similar tjStuff+ in consecutive seasons, which supports the use of tjStuff+ as a predictive statistic.

Updating the Model

I am content with how the model is performing on a predictive level. To evaluate the model on a descriptive level, I will retrain the model using 2020–23 data, and then test on 2024 data.

I will use the same methodology above, only now I will include 2023 data.

tjStuff+ Benchmarks

We have discuseed that tjStuff+ is a metric which predicts the expected run value (xRV) of a given pitch by its physical characteristics.

To better understand tjStuff+ it is imparative that we take a look at what defines the metrics distribution. The following metrics define the normal distributon of xRV.

Expected Run Value Metrics

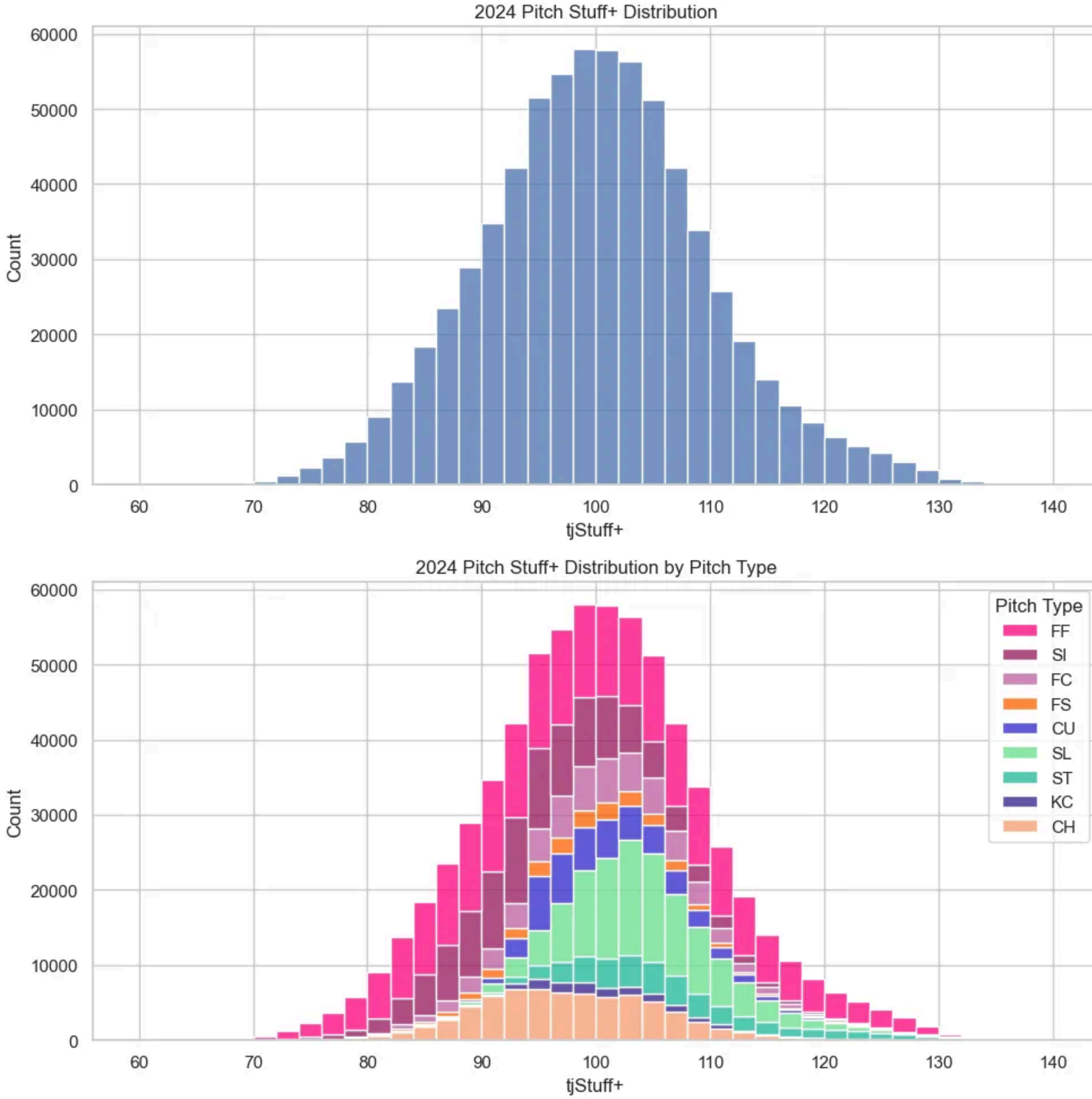
- Mean xRV/100: 0.35 (positive means favours batter)
- StDev xRV/100: 0.68

Let’s do an example working backwards from a tjStuff+ Value:

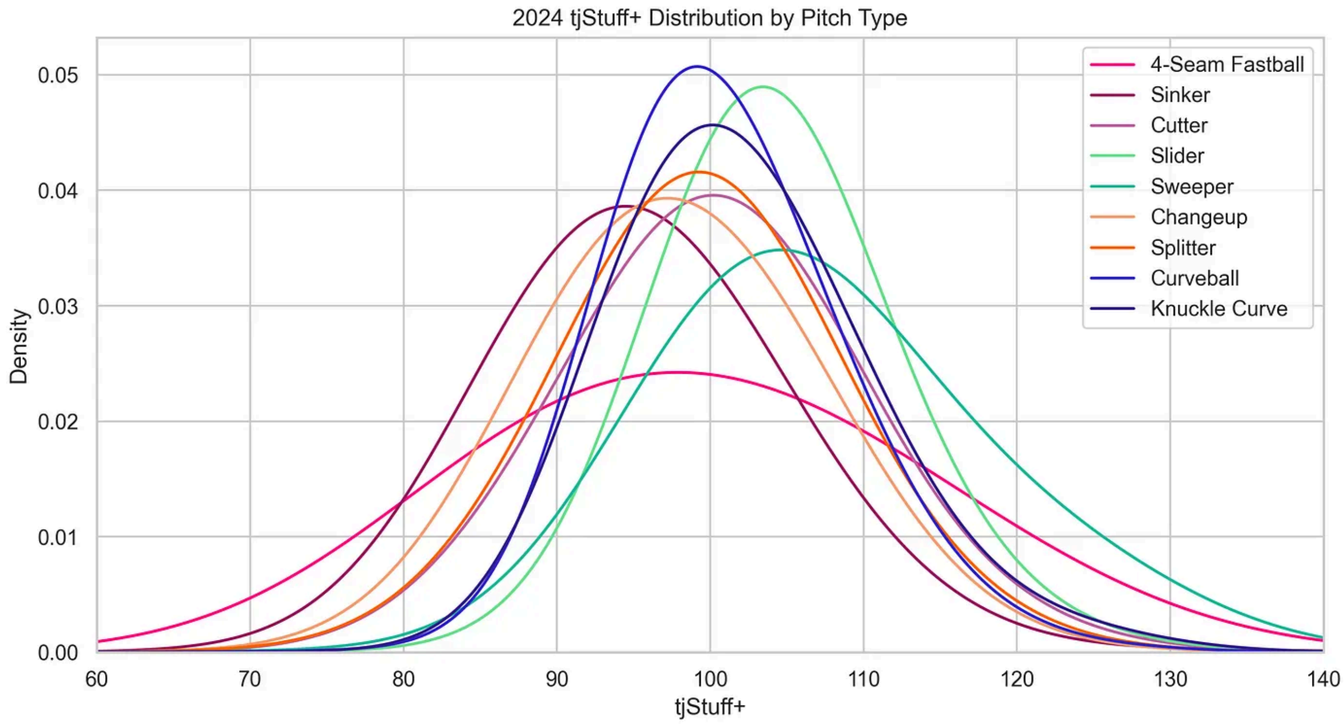
> *Assume you have a pitch which has a tjStuff+ value of 130. What this means is that the pitch is -3 Standard Deviations below the mean (we inverse xRV becasue positive xRV favours batters).*
>
> *Working backwards we, know that 1σ = 0.68 xRV/100, so 3σ = 2.04 xRV/100*
>
> *When the model outputs a value of 130 tjStuff+ it is computing that the pitch at a 100 Pitch Rate provides the pitcher with +2 Run Value compared to the average pitch*

Validation

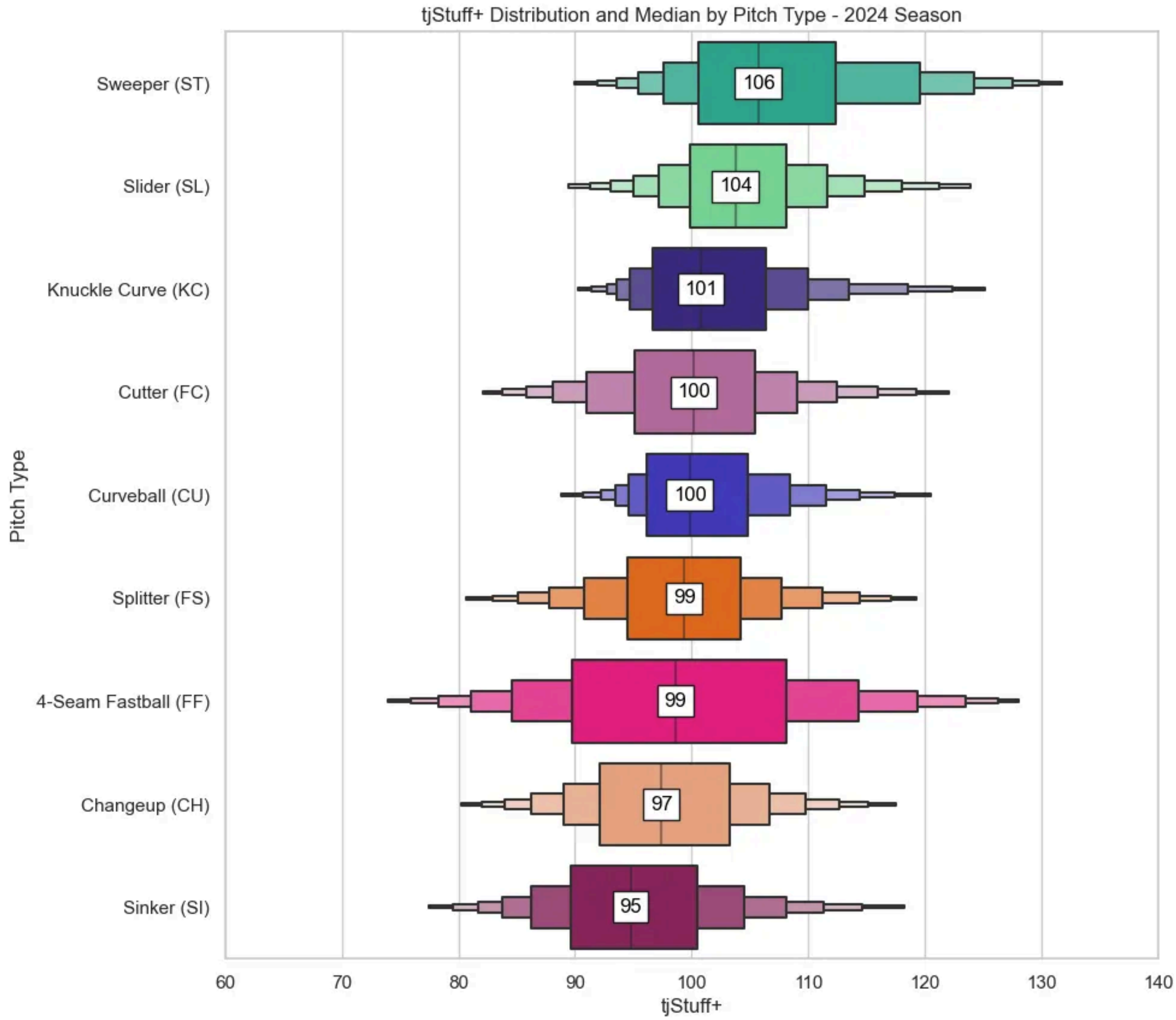
Let’s take a look 2024 tjStuff+ Distribution and Median by All pitches and then Pitch Type.



2024 Pitch Level tjStuff+

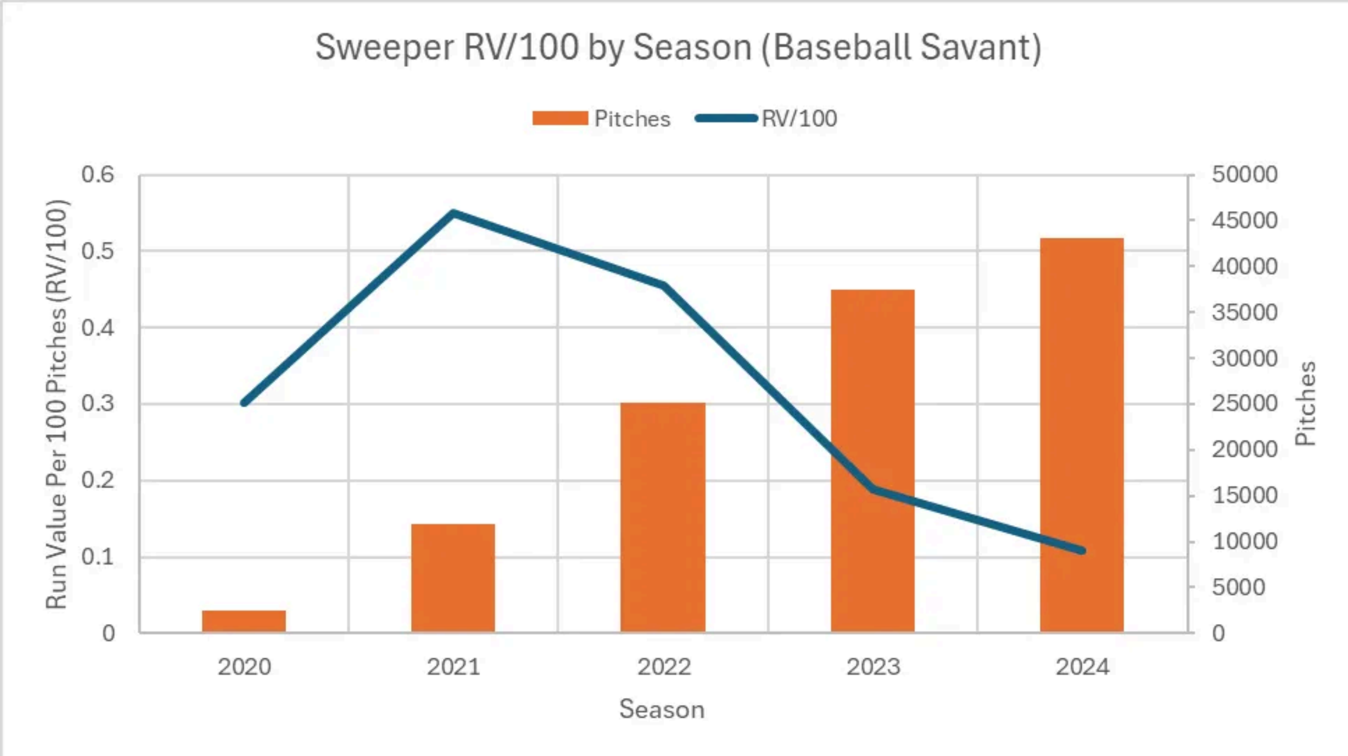


2024 tjStuff+ Distribution by Pitch Type



2024 Pitch Type tjStuff+ Boxen Plots

It is interesting to note that the distribution of sweepers tightened and shifted downwards from the 2023 metrics. Including the 2023 seems to has decreased sweeper grades a whole, and it makes total sense! Sweepers grew in popularity during the 2021 season and remained the most effective pitches in baseball throughout the 2022 season. It wasn't until the 2023 season where sweepers started to decline in effectiveness, and this carried into 2024. This decline was likely caused by batters becoming more familiar with the offering and that many pitchers started throwing the offering, which meant that there were more "poor" sweepers being thrown.



Sweeper Run Value Plot

Since we trained off 2020–2022 for the first iteration, we did not capture the decline of sweepers that occurred in the 2023 season. Including these sweepers is the main reason for the tjStuff+ differential (and is the only instance where I use a plot from Excel).

Descriptiveness

We are left with assessing the descriptiveness of the model as we can only compare 2024 Results. The following table displays the correlation between our specified metrics during the 2024 season.

```
Minimum Pitches: 100
Sample Size: 711
Average Pitches: 989
Correlation between 2024 and 2024:
```

	2024 tjStuff+	2024 FIP	2024 wOBA	2024 K-BB%
2024 tjStuff+	1.00	0.28	0.30	0.40
2024 FIP	0.28	1.00	0.79	0.66
2024 wOBA	0.30	0.79	1.00	0.54
2024 K-BB%	0.40	0.66	0.54	1.00

2024 vs 2024 Correlation Matrix

tjStuff+ is not designed to be a descriptive metric, and as such, it does not perform well as one. More conventional metrics like FIP and K-BB% are superior in terms of describing a pitchers' performance. tjStuff+ lack of descriptive power can be heavily explained by its lack of location information.

2024 Metrics

Player Metrics

Pitch Grades

With the model trained and validated, we can now apply it to 2024 data to get all sorts of metrics! Let’s take a look at tjStuff+ by pitcher and pitch type and create a leaderboard.

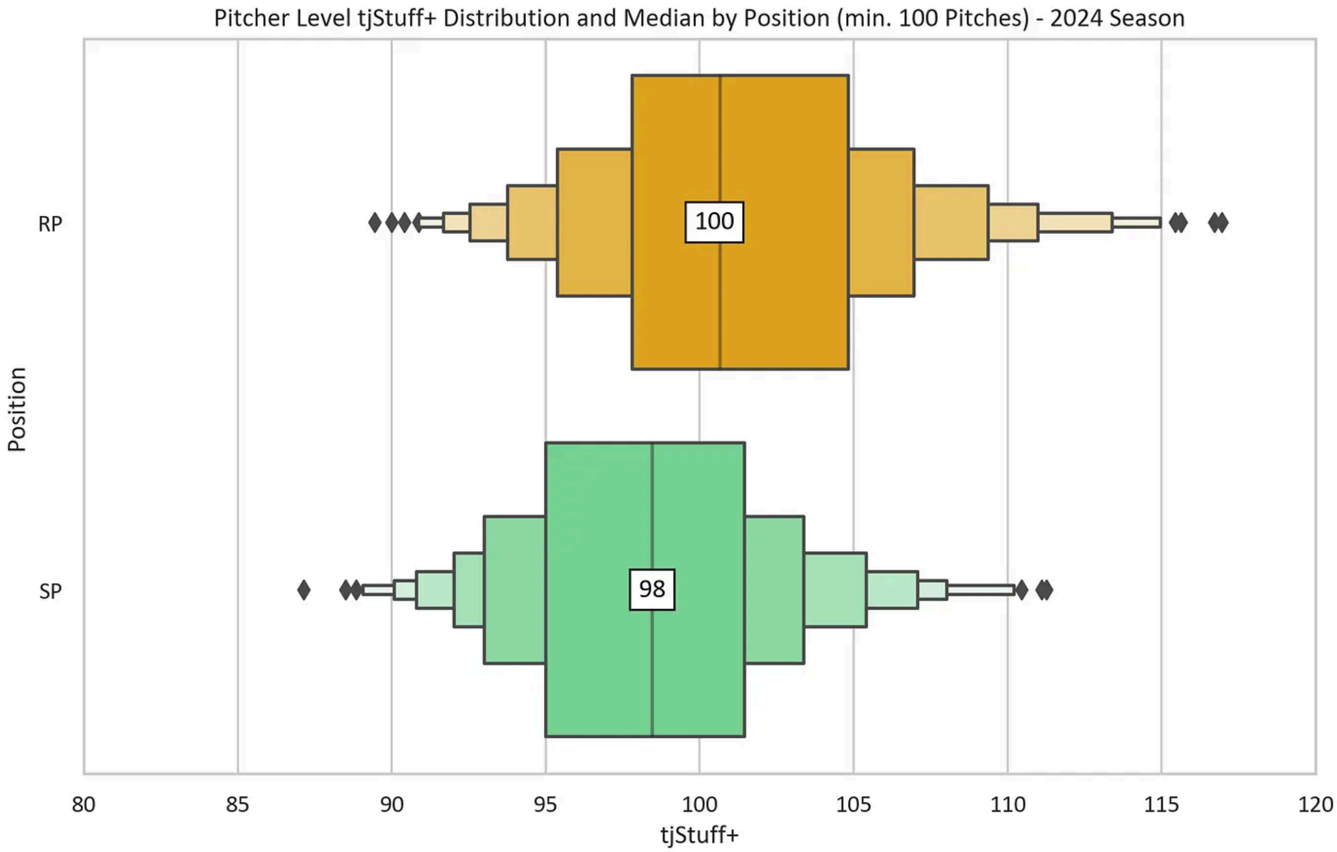
To better contextualize tjStuff+, I also calculate a ‘Pitch Grade’ for each pitch type which is scaled to the traditional 20–80 Scouting Grades. It is normally distributed, however the Standard Deviation (σ) is determined by taking the difference between the 99.9th and 0.1th Percentile of tjStuff+. This ensures that the greatest tjStuff+ pitch of a specific type is graded at 80, while the worst tjStuff+ pitch is graded at 20.

I decided to make it like this because applying the Standard deviation at the pitch level for each pitch type caused very tight distributions, especially for 4-Seam Fastballs. The greatest 4-Seam "Pitch Grade" for this method was 65. While it is mathematically sound, having the best Fastballs in baseball graded as "Good" rather than "Elite" did not sit well with me. Each pitch type has pitches that can span from 20 to 80 in grade, with grades following a normal distribution.

Startes vs Relivers

Let's take a look at tjStuff+ by position. Starters (SP) and Relievers (RP) play two distinct roles in baseball. Starters are tasked with pitching longer outings and are geared towards command and control rather than higher velocity and strikeout numbers. Relievers are quite the opposite, as they pitch shorter outings and tend to post incredible K% with less emphasis on lower BB%.

This shows up in the distribution of tjStuff+ by SP and RP. SP are more clustered together with just a handful displaying elite stuff, while RP is positively skewed. Thanks to their shorter outing, RP can consistently output higher quality pitches, making both the average and the max greater than SP.



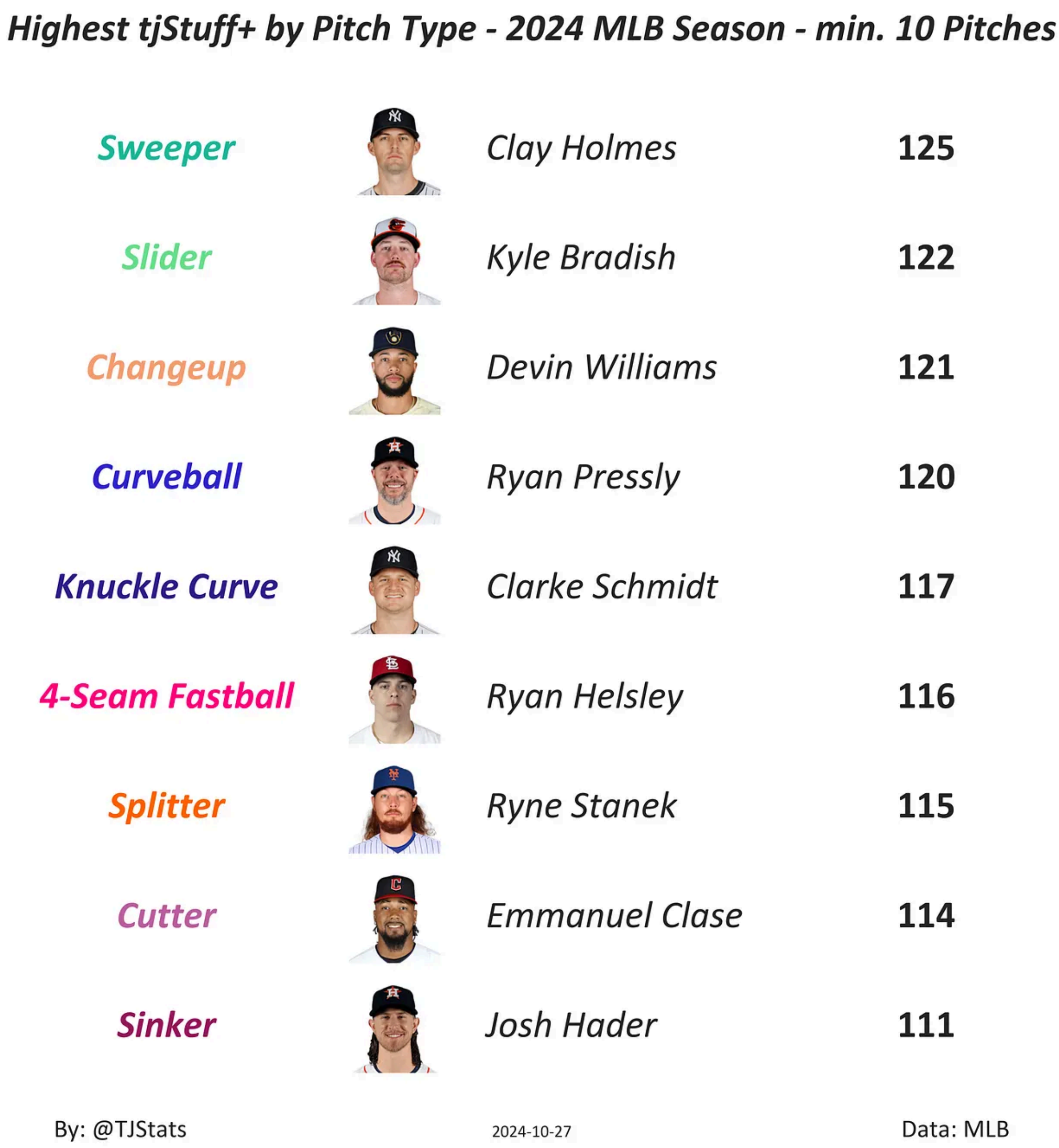
2024 Pitcher Level tjStuff+ by Position

tjStuff+ Leaders

The following graphic is a simple leader board of the best tjStuff+ pitchers during the 2024 MLB Season. It could be displayed in a table, but I like illustrating leader boards in different ways, such as this.

2024 tjStuff+ Leaders

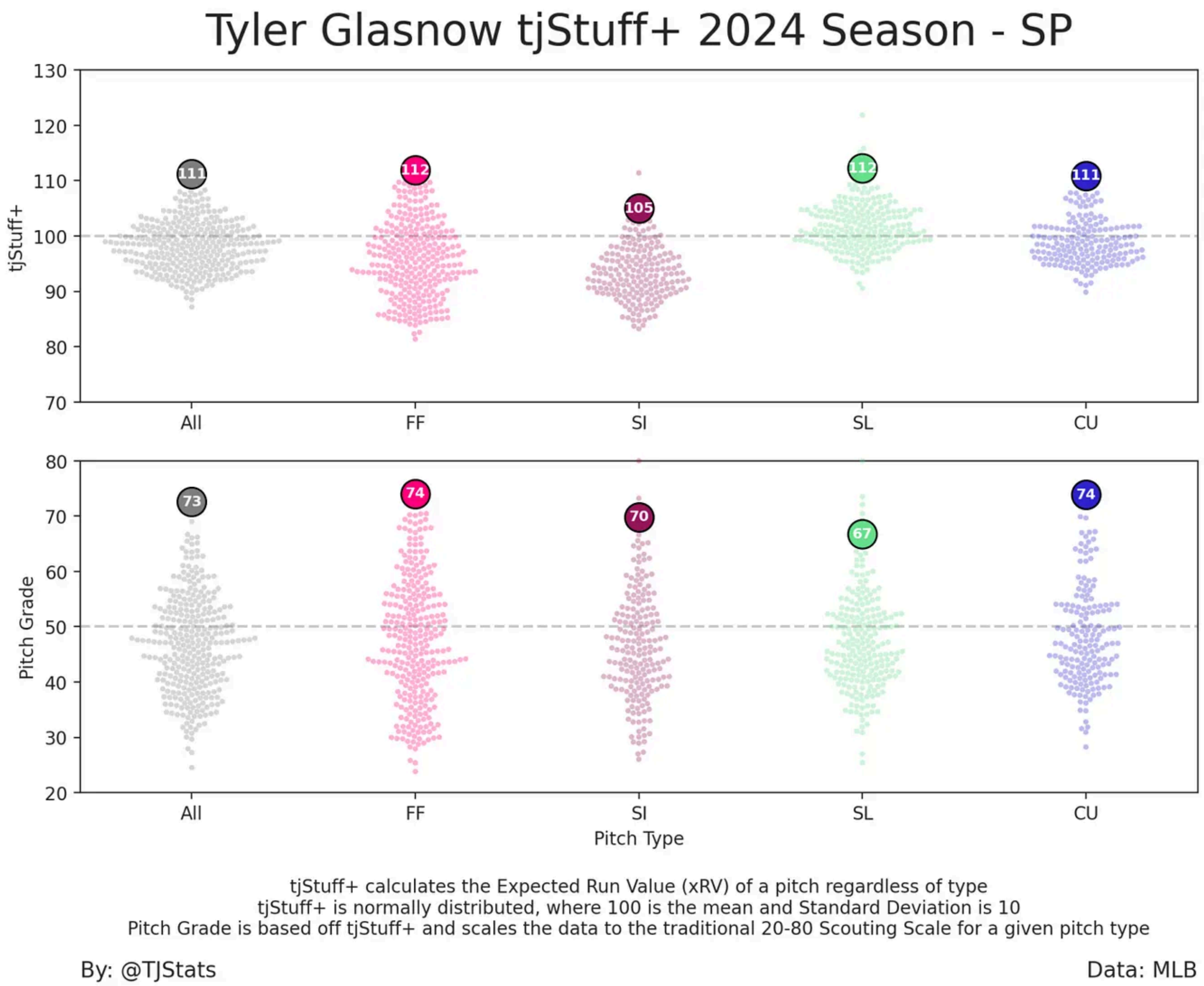
This graphic is a leader board of the greatest tjStuff+ by Pitch Type. These pitches are assigned a "Pitch Grade" of 80 by our aforementioned definition.



2024 tjStuff+ Pitch Type Leaders

Pitcher Summary

I created a [Streamlit app]() which tabulates and plots tjStuff+, for all MLB players during the 2024 MLB Season. Here is an example of one of the plots.

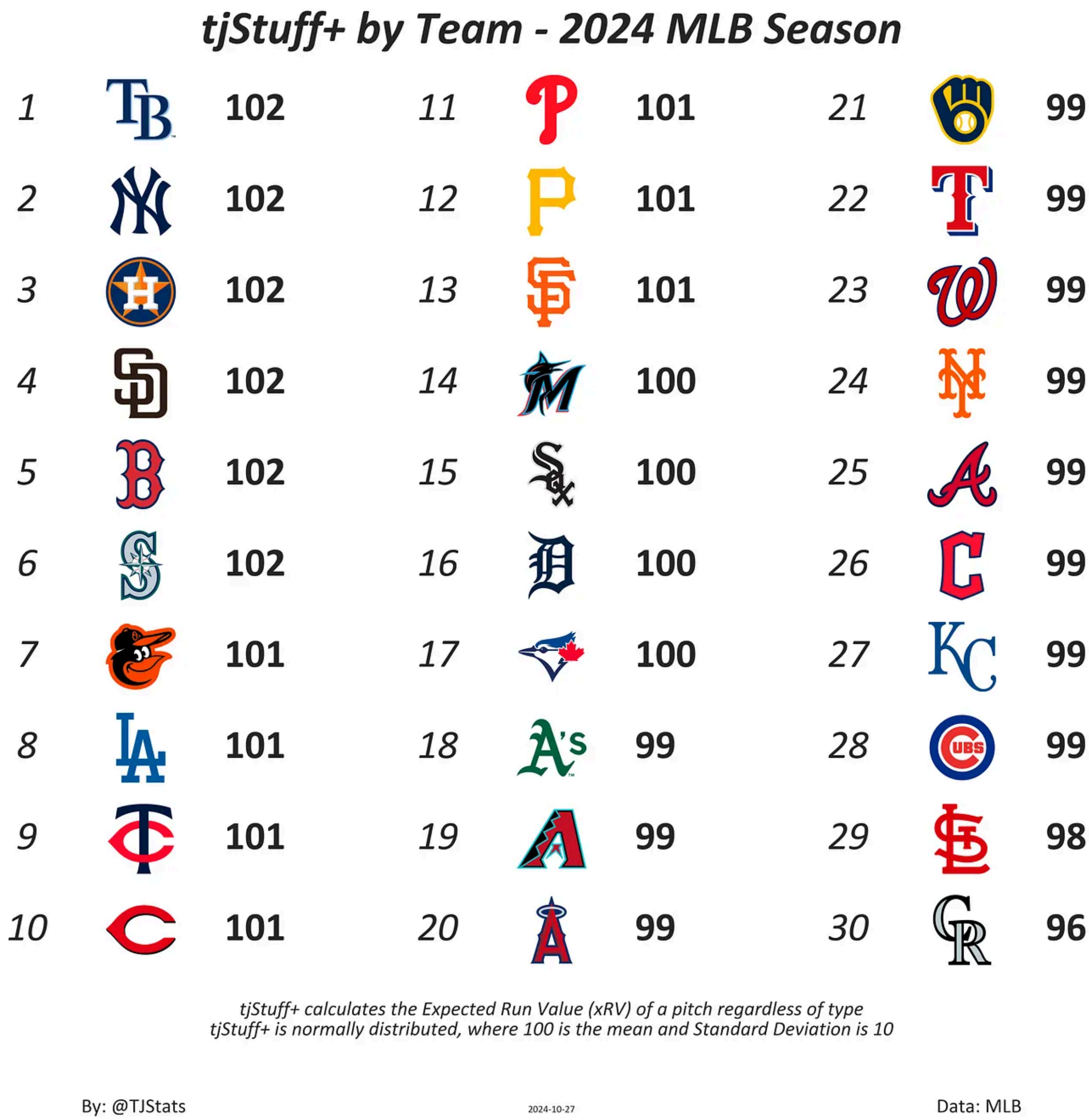


Tyler Glasnow 2024 tjStuff+ Summary

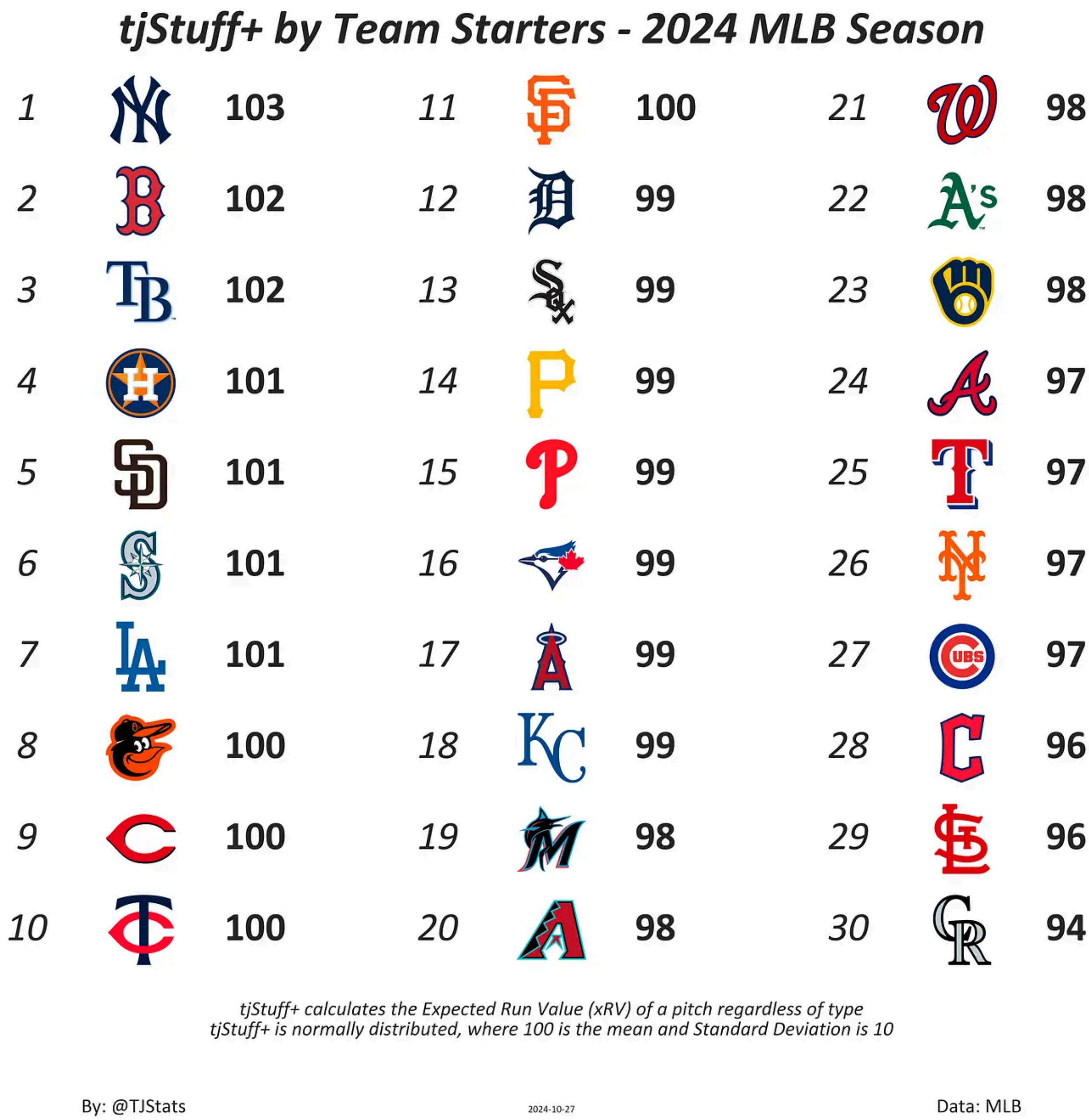
Team Metrics

Let's calculate some team metrics!

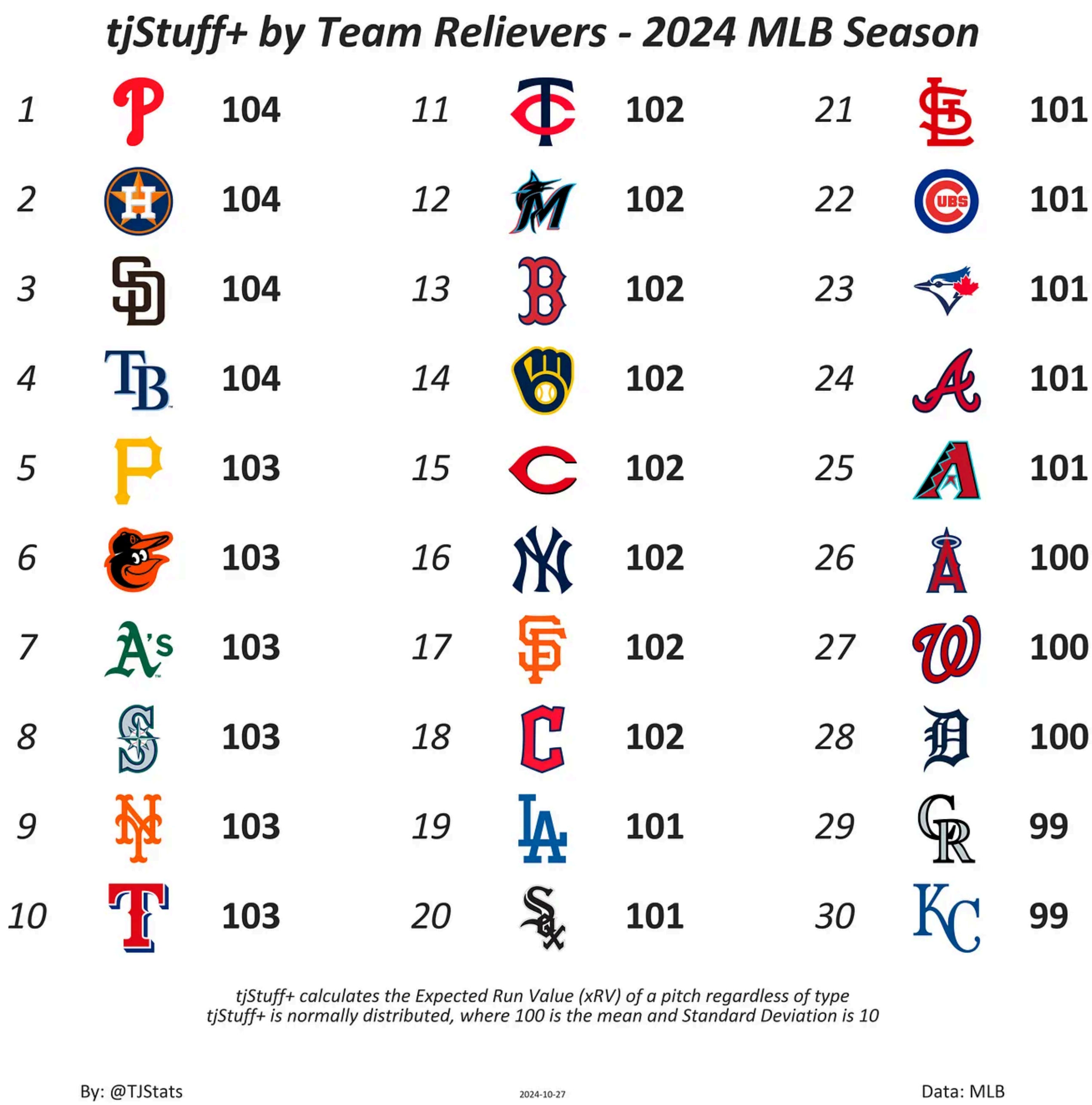
Here is a leader board for tjStuff+ by team. I will also show tjStuff+ for Starters and Relievers.



2024 Team Level tjStuff+



2024 Team Starters Level tjStuff+



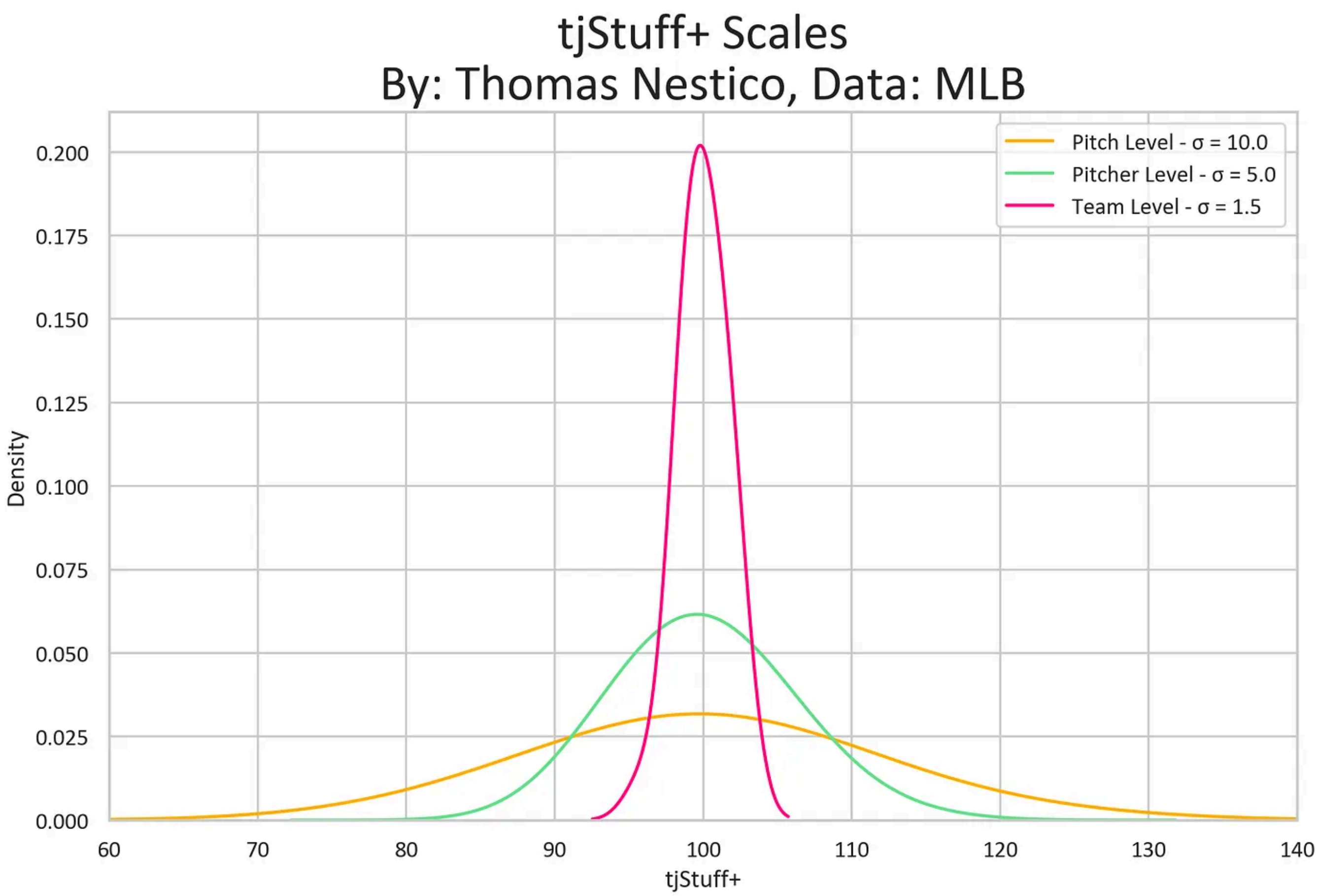
2024 Team Relievers Level tjStuff+

Scaling

We can look at the distribution of tjStuff+ as we aggregate to different levels. Recall, tjStuff+ is normally distributed with a mean of 100 and a standard deviation of 10 at the pitch level. As we aggregate, we deal with larger and larger samples which regresses tjStuff+ to the mean. I do not scale tjStuff+

after aggregation, so it is important to understand how the distribution of tjStuff+ varies at different aggregation levels.

The following plot illustrates how the distribution of tjStuff+ tighten as we aggregate.



tjStuff+ Scales

Park Factors

The physical characteristics of a pitch can vary depending on the environment. The most popular example of this is Coors Field in Colorado which is notorious for its extreme elevation, sitting 5200 ft above seas level. Due to this elevation, the air is less dense Colorado, which causes pitches to have less overall movement. This causes pitches in Colorado to be negatively affected in the calculation of tjStuff+.

The way I calculate the park factors is first computing each teams tjStuff+ at home and on the road. After than I transform the tjStuff+ values into respective Z-Score and then Calculate the CDF probability. Finally, I divide Home CDF by Road CDF and multiply by 100 to get the Park Factor.

2024 tjStuff+ Park Factors

Conclusion

Creating my own pitching model has taught me a lot about both baseball and data analytics. It has also allowed me to flex my programming skills while picking up new tools such as Polars as Streamlit.

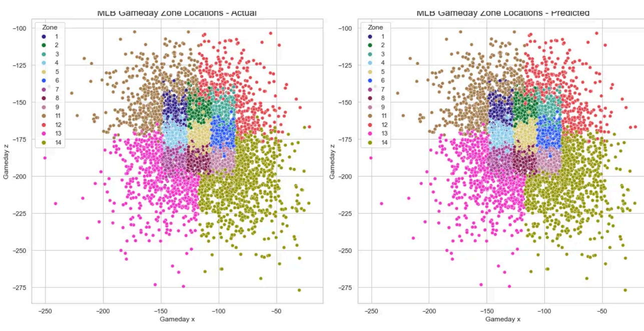
I hope you enjoyed my journey on updating my tjStuff+ mode and that it inspires you to tackle others projects you are interested in.

Thank you for reading!

Follow me on Twitter: https://x.com/TJStats

All code for this project is available on GitHub

More from Thomas Nestico

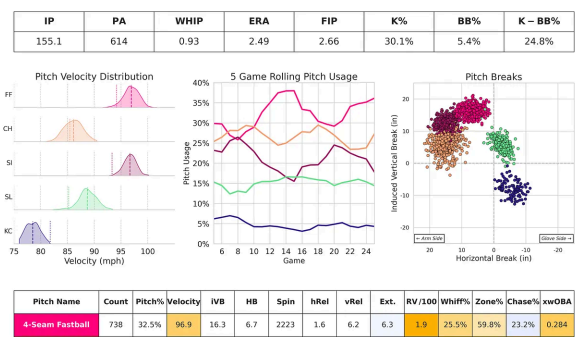


Thomas Nestico

Classifying MLB Pitch Zones and Predicting MiLB Zones

Statcast Pitch Classification

Nov 17, 2023 9

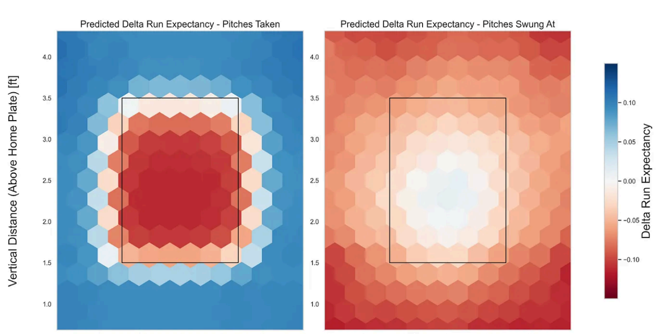


Thomas Nestico

Creating the Perfect Pitching Summary

Introduction

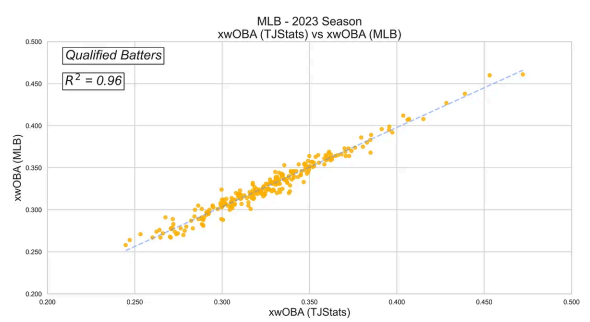
Aug 22, 2024 18 2



Thomas Nestico

Modelling Batter Decision Value

Introduction



Thomas Nestico

Modelling xwOBA (With KNN)

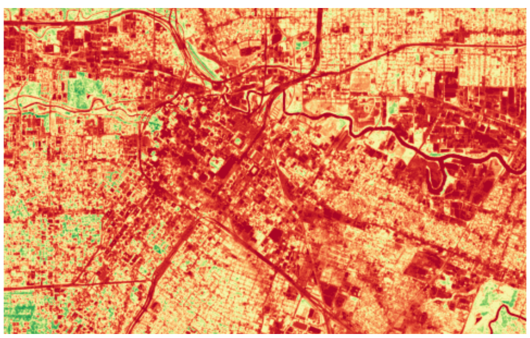
Introduction

Dec 1, 2023 11

Feb 13, 2024 54 1

See all from Thomas Nestico

Recommended from Medium



Kyle Jones

Tracking Houston's Green Cover with Satellite NDVI using Remote…

Change often happens in ways we can't quite see—until we pull back far enough. From hig…

Apr 21 81

Gayatri Gattani

How Do Airlines Forecast Ticket Demand? (Easy way to understan…

Imagine you're on the data science team at Delta Airlines.

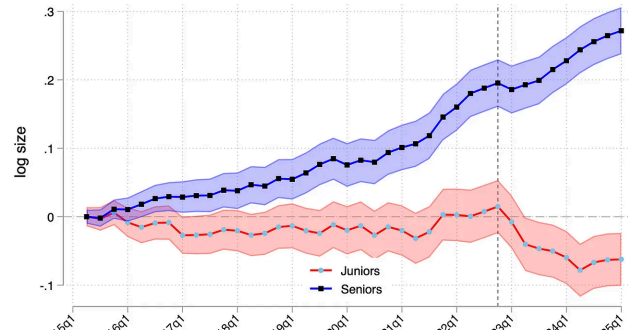
Sep 1 22 2

In Long. Sweet. Valuable. by Ossai Chinedum

I’ll Instantly Know You Used Chat Gpt If I See This

Trust me you’re not as slick as you think

May 16 · 24K · 1433

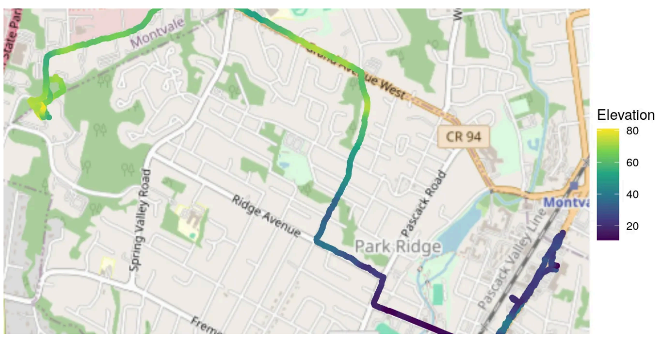


In Data Science Collective by Andres Vourakis

AI and the Data Science Job Market: What the Hell Is Actually…

What aspiring, junior, and senior data scientists should know to stay future-proof

Sep 16 · 981 · 39



Brian Carey

GPS Mapping with R

A comparison with Python

Mar 29 · 1

Nicole Venglovicova

Hockey Champions League Predictions — 7 September 2025

Get the stats, picks, predictions and analysis

Sep 6 · 29

See more recommendations