

# From Rookie to Cy Young: Evaluating the Performance Evolution of MLB Pitcher Corbin Burnes from 2019-2023

STA 160 Final Project

Sid Das, Vincent Barletta, Martin Topacio, Felix Yang

2024-06-11

## Introduction

MLB (Major League Baseball), since the advent of big data, has proven to be ripe for analysis due to the many factors and intricacies within the game, best exemplified by its advanced metrics. In this report, we investigate MLB pitcher Corbin Burnes and dive into his evolution as a pitcher over a five-year span of 2019-2023, taking a look at some of the advanced metrics behind his pitches. Corbin Burnes was our selection due to him having a very interesting career, especially over his last five seasons. Burnes struggled mightily after making his MLB debut in 2018, with the struggles continuing in 2019 before he found success and dominance in 2020, finishing 6th in NL (National League) Cy Young voting, the highest honor a pitcher can achieve. His dominance continued in 2021 and to the present, with our interval for this report covering his 2019 through 2023 seasons.

Through analysis of Burnes' pitches, we plan to create a "good" or "bad" pitch function based on run expectancy, which is the average number of runs an average team is expected to score during an inning. This metric works to credit a pitcher's role in not allowing a run in a given inning. With our function, we can make inferences on Burnes' performance as a pitcher, which revolves around his ability to prevent runs.

Related to our analysis of his performance are the metrics we use to evaluate and infer. We chose the advanced metrics of spin rate, batting average, wOBA, and effective speed. Spin rate is measured by rotations per second, with breaking pitches like curveballs and sliders having the most break. Fastballs and its variations, like cutters and sinkers, also have spin, just less so than breaking balls. Generally, the most effective pitches of any variation have a lot of spin or break on it. Batting average is measured by the rate at which the pitcher allows a hit of any kind, whether it is a single, double, triple, or home run.

wOBA is a variation of on-base percentage, which is the rate at which a pitcher allows the hitter to reach base, whether it is from any kind of hit, a walk, or a hit-by-pitch. The difference with wOBA is how it factors how a player reaches base, giving more weight to extra-base hits. Effective speed is how fast the ball is tracked when it crosses home plate, with fastballs and its variations being faster pitches generally than breaking pitches. With these metrics, we can delve into how Burnes may have achieved his success from 2020-2023 as we compare and contrast his metrics from that span to his 2019 season. Additionally, with our "good"/"bad" pitches function, we will be able to see what his metrics were at the pitches that achieved the most and least success, giving us an idea as to why he was and was not successful in those instances.

## Exploratory Data Analysis

First, we will begin by creating our webscraping function to gather all of our necessary data from MLB Statcast.

### All Pitches by Burnes (2019-2023)

```
## -- MLB Baseball Savant Statcast Search data from baseballsavant.mlb.com -----
```

```
## i Data updated: 2024-06-11 20:00:10 PDT
```

```
## # A tibble: 6 x 95
##   pitch_type game_date release_~1 relea~2 relea~3 playe~4 batter pitcher events
##   <chr>      <date>      <dbl> <dbl> <dbl> <chr>    <dbl> <dbl> <chr>
## 1 FF        2019-03-31      96    -1.42  5.89 Burnes~ 542303 669203 "fiel~
## 2 SL        2019-03-31     85.5   -1.45  5.77 Burnes~ 500874 669203 "stri~
## 3 FF        2019-03-31     94.4   -1.37  5.85 Burnes~ 500874 669203 ""
## 4 SL        2019-03-31     89.6   -1.55  5.79 Burnes~ 500874 669203 ""
## 5 SL        2019-03-31     88.8   -1.48  5.81 Burnes~ 500874 669203 ""
## 6 SL        2019-03-31     86.8   -1.41  5.8  Burnes~ 500874 669203 ""
## # ... with 86 more variables: description <chr>, spin_dir <lgl>,
## #   spin_rate_deprecated <lgl>, break_angle_deprecated <lgl>,
## #   break_length_deprecated <lgl>, zone <dbl>, des <chr>, game_type <chr>,
## #   stand <chr>, p_throws <chr>, home_team <chr>, away_team <chr>, type <chr>,
## #   hit_location <int>, bb_type <chr>, balls <int>, strikes <int>,
## #   game_year <int>, pfx_x <dbl>, pfx_z <dbl>, plate_x <dbl>, plate_z <dbl>,
## #   on_3b <dbl>, on_2b <dbl>, on_1b <dbl>, outs_when_up <int>, ...
```

Displayed is a small tibble of Corbin Burnes' pitches in every game played from 2019 to 2023, with metrics of release speed and run expectancy accompanied.

### **Burnes' Average Spin Rate, Estimated wOBA, Estimated Batting Average, and Effective Speed Per Game**

This data frame displays the value of Corbin Burnes' average effective speed, estimated wOBA, estimated batting average, and average spin rate in every single game he has played in, with the year and value for each metric.

```
##   game_date      variable      value Year
## 1 2019-03-31 Average Effective Speed 92.18046 2019
## 2 2019-04-06 Average Effective Speed 92.52473 2019
## 3 2019-04-12 Average Effective Speed 93.10957 2019
## 4 2019-04-17 Average Effective Speed 91.63649 2019
## 5 2019-05-01 Average Effective Speed 91.23750 2019
## 6 2019-05-04 Average Effective Speed 93.91935 2019
```

### **Burnes' Total Average Spin Rate, Estimated wOBA, Estimated Batting Average, and Effective Speed Per Year**

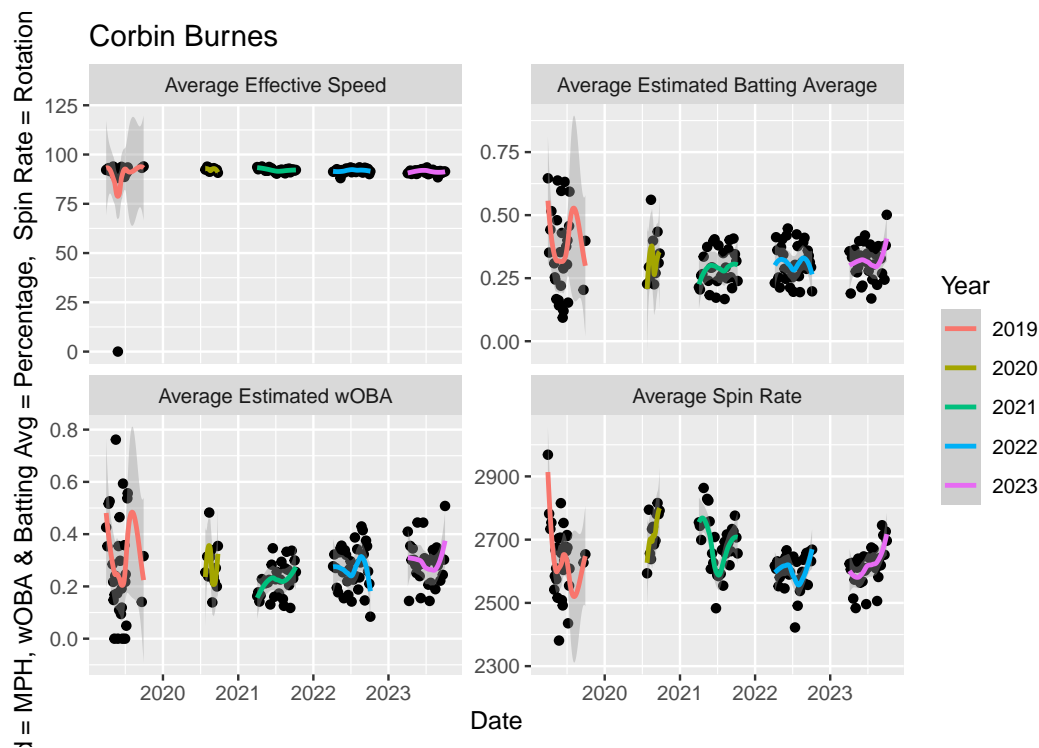
For every year between 2019 and 2023, we obtain the total averages of Corbin Burnes' average spin rate, estimated wOBA, estimated batting average, and effective speed. The total averages are displayed in the 5x5 tibble.

```
## # A tibble: 5 x 5
##   Year Estimated_Batting_Average Effective_Speed Estimated_wOBA Spin_Rate
##   <fct>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 2019          0.412          90.2          0.358      2692.
## 2 2020          0.321          92.4          0.273      2729.
## 3 2021          0.288          92.2          0.22       2686.
## 4 2022          0.317          91.9          0.273      2601.
## 5 2023          0.318          91.3          0.29      2619.
```

# Visualizing Burnes' Pitching Progress and Prowess

## Graph of Average Pitching Stats from 2019-2023

Four graphs are displayed, for each metric discussed in the previous table. Effective speed is presented as miles per hour (MPH), wOBA and batting average are presented as a percentage, and spin rate is presented as rotations per second. These metrics are tracked over each of the five years.



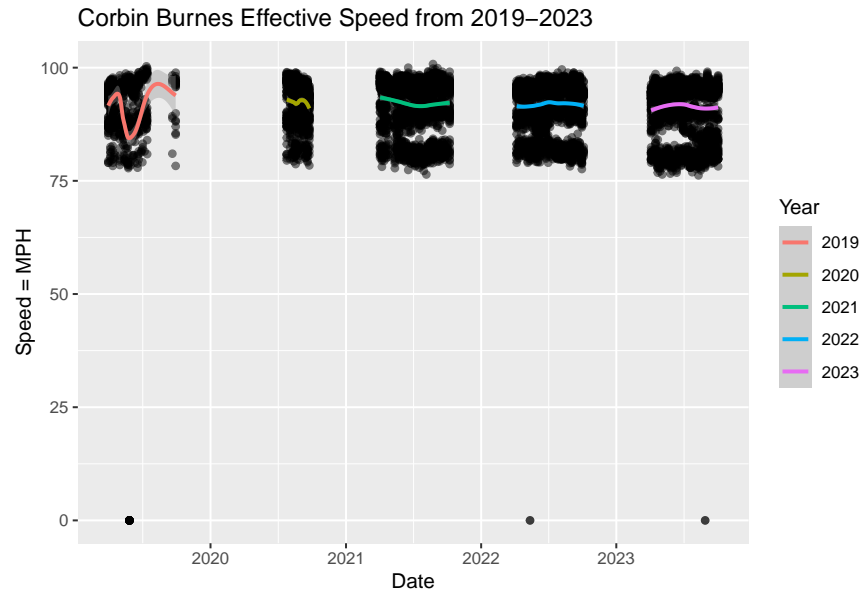
Of note, effective speed had the highest variance in 2019, possibly indicating how Burnes may have thrown more pitches of different type and speed that year. Examining effective speed for the next four years, the lines are much more straight than 2019's, indicating that Burnes did not mix his pitches as much.

For estimated batting average, the graph tracks Burnes' opponents' batting averages off of his pitches for every start, and we see for 2019 how varied his estimated batting average is. The line graph reaches a peak higher than any of the other years, and its low is still over .250, which would indicate that opponents had great success getting hits off Burnes in 2019. The lines are much smaller and tighter from 2021 to 2023, sitting lower than 2019. This indicates that Burnes had much more success preventing hits from 2021 to 2023.

Further, we examine the wOBA graph, which is similar to batting average, but as a metric accounts more for how a batter reaches base, i.e. from walks or extra base hits. Like the estimated batting average graph, the estimated wOBA is much less clean for Burnes in 2019. His estimated wOBA from 2020 to 2023 has less variance, with his 2021 and 2022 seasons in particular having tighter distributions. Lastly, we see from Burnes' spin rate graphs that his spin rate in 2019 is all over the place. A possible and logical explanation for this can be connecting it to his effective speed also having high variance. Burnes had a more varied pitch mix in 2019, leading to many different speeds and spin rates due to using different types of pitches more often. It is well-documented that Burnes switched to using his cutter pitch far more frequently in his seasons after 2019, which would explain the tighter distributions in the years following 2019.

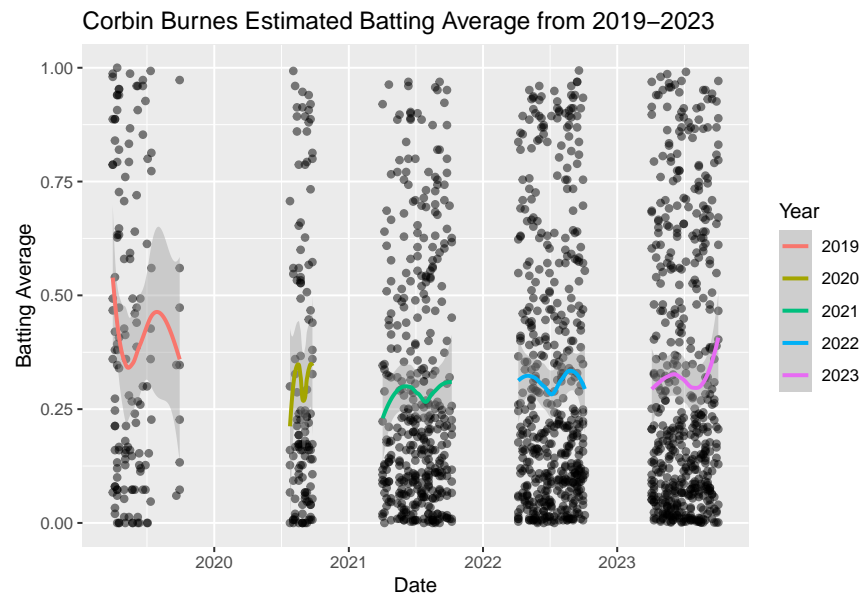
## Graphs of Raw Stats from 2019-2023

### Effective Speed:



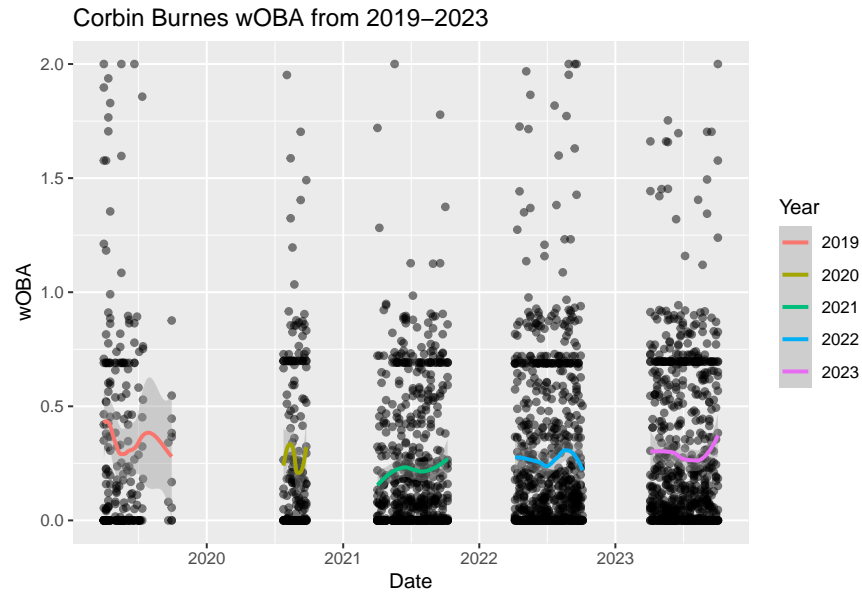
We again see the line graph for 2019 being more varied compared to 2020 to 2023, as a result of Burnes using a more varied pitch mix in 2019. We can also infer from the graphs from 2020 to 2023 that in 2020, when Burnes started using less pitches and choosing the cutter more often, that the line begins to get tighter and smoother with every year, most likely as a result of Burnes becoming more comfortable with his pitch mix, specifically his cutter.

#### Estimated Batting Average:



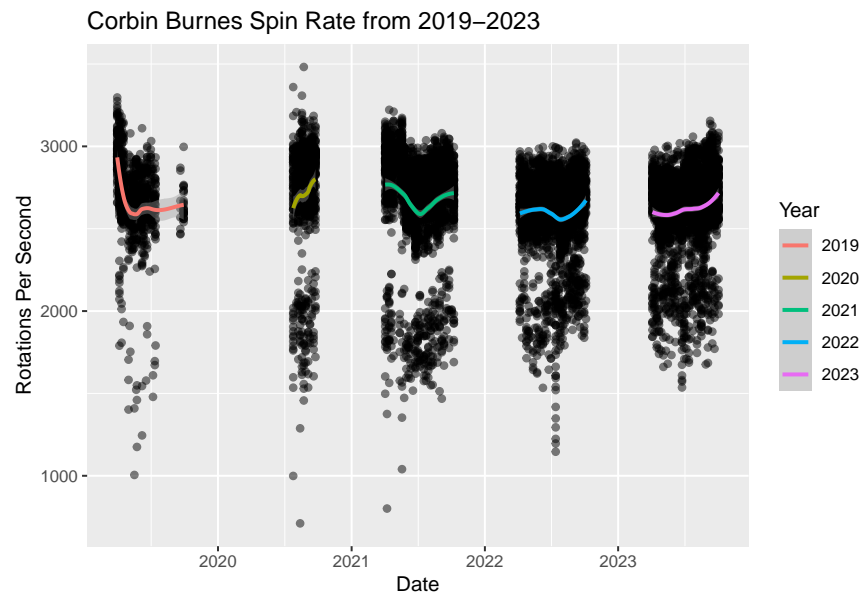
Burnes' line graph for his estimated batting average in 2019 sits at a level higher than the other years, the reasons for which we have discussed in the previous section. Interestingly, we can take a look at 2023 compared to 2021 and 2022 and infer that Burnes struggled in 2023 relative to how dominant he was in 2021 and 2022.

#### Estimated wOBA:



Again, from these graphs we get a similar inference as estimated batting average, and we can see that Burnes struggled mightily in 2019, found his footing from 2020-2023, but also struggled relative to the new standard he had set in 2023.

### Spin Rate:



We can get a closer look at the spin rates per game for Burnes in our five-year span, seeing that in 2019, his spin rate took a steep drop in the middle of the season, likely a result of Burnes experimenting with his pitch mix. We see that Burnes' spin rate slowly increased over the course of the COVID-19 shortened season in 2020. 2021 had Burnes roughly emulating an upside-down bell curve with his spin rate, and he has comparatively cleaner average spin rates in 2022 and 2023.

## Diving Deeper: Burnes' Good Pitches vs. Bad Pitches

We filter Burnes' pitches by either "good" pitches or "bad" pitches. We define "good" pitches to have a run expectancy below 0, and a "bad" pitch to have a run expectancy above 0. Essentially, after a pitch is thrown, a bad pitch will cause the probability of a batter scoring to increase while a good pitch will lower the probability of scoring.

We want to portray how high Burnes' highs are as a pitcher and how low his lows are here. We know that Burnes achieved great success post-2019 season, but we want to see how his best and worst starts (with every pitch) stack up to each other.

### Burnes' Average Spin Rate, Estimated wOBA, Estimated Batting Average, and Effective Speed Per Year

```
## # A tibble: 5 x 5
##   Year Estimated_Batting_Average Effective_Speed Estimated_wOBA Spin_Rate
##   <fct>                <dbl>          <dbl>          <dbl>    <dbl>
## 1 2019                0.201            89.9            0.098    2706.
## 2 2020                0.19             92.5            0.116    2758.
## 3 2021                0.184            92.3            0.104    2709.
## 4 2022                0.192            92.1            0.115    2602.
## 5 2023                0.199            91.7            0.134    2619.

## # A tibble: 5 x 5
##   Year Estimated_Batting_Average Effective_Speed Estimated_wOBA Spin_Rate
##   <fct>                <dbl>          <dbl>          <dbl>    <dbl>
## 1 2019                0.614            90.3            0.745    2676.
## 2 2020                0.619            92.2            0.687    2690.
## 3 2021                0.503            91.9            0.555    2645.
## 4 2022                0.595            91.6            0.693    2599.
## 5 2023                0.617            90.9            0.694    2614.
```

Comparing the tables, we can highlight how in his 2019 season, his estimated wOBA is both the highest and lowest for good and bad pitches across both tables. His peak .098 wOBA is the lowest mark for good pitches, and his low .745 wOBA is the highest in the bad pitches table. We can also note that while these tables may portray Burnes to have similar seasons for 5 straight years, the 2019 season features far more "bad" pitches than "good" ones, and that our pitch system only highlights him on a per-start basis based on the good/bad.

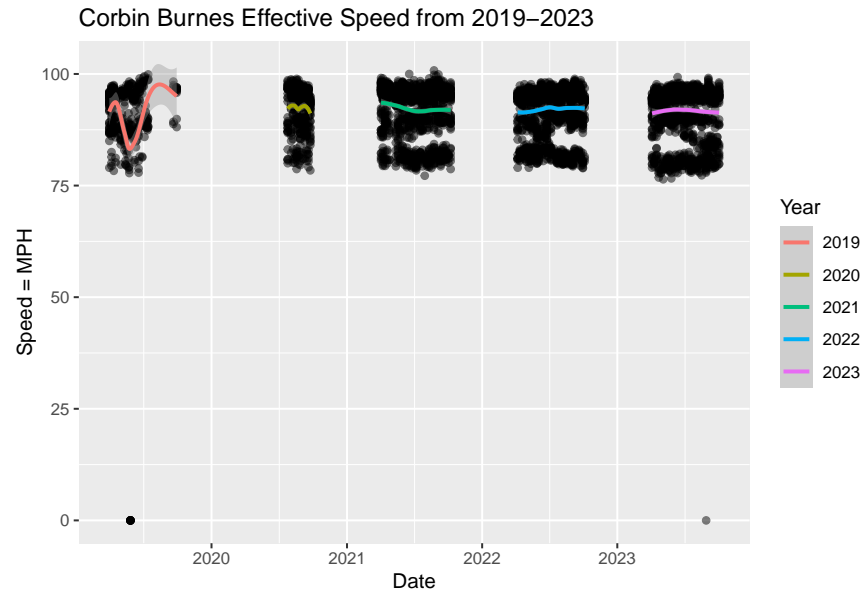
## Graph of Burnes' Good Pitches from Average Stats



These graphs illustrate Burnes' average metrics from only his good pitches. What we can obtain from examining his graph for effective speed in 2019 is the variance in what his good pitches were, once again owing to his more varied pitch mix. Once Burnes fell in love with his cutter after this season, we see that the effective speed graphs from 2020-2023 are much more straight, indicating that his good pitches were of similar speeds around 90 MPH, indicating some type of fastball being thrown as a good pitch (his cutter). From the estimated batting average graph, we can see the inconsistency in Burnes during his 2019 season, with even good pitches resulting in an estimated batting average of close to .300, which is not good. Looking at his 2021-2023 seasons, the batting average graph has less variance and hovers around .200, which indicates how effective his good pitches were. For wOBA, it's a similar story with 2021-2023 being much cleaner, but we can see how 2020 was also up-and-down like 2019, albeit with a smaller sample size. The spin rate for 2019 peaks at a level much higher than the other years, due to Burnes likely choosing to use more breaking pitches (more spin). We can see how Burnes commits more to his good pitches having less spin in 2022 and 2023.

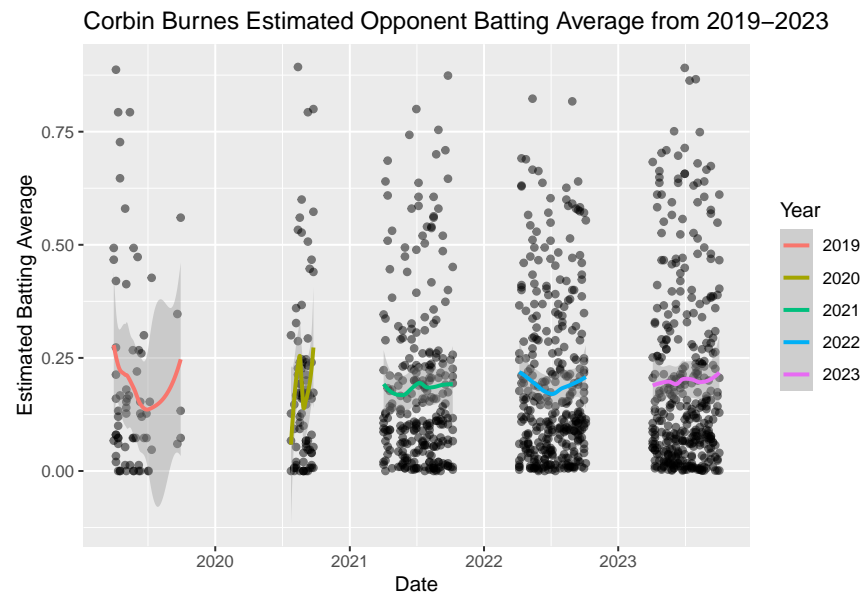
## Graphs of Good Pitches from Raw Stats

### Effective Speed:



Across every pitch Burnes throws over our five-year interval, we can again see how the speed of his good pitches varied heavily in 2019, varied a little across a small sample size in 2020, and straightened out from 2021-2023.

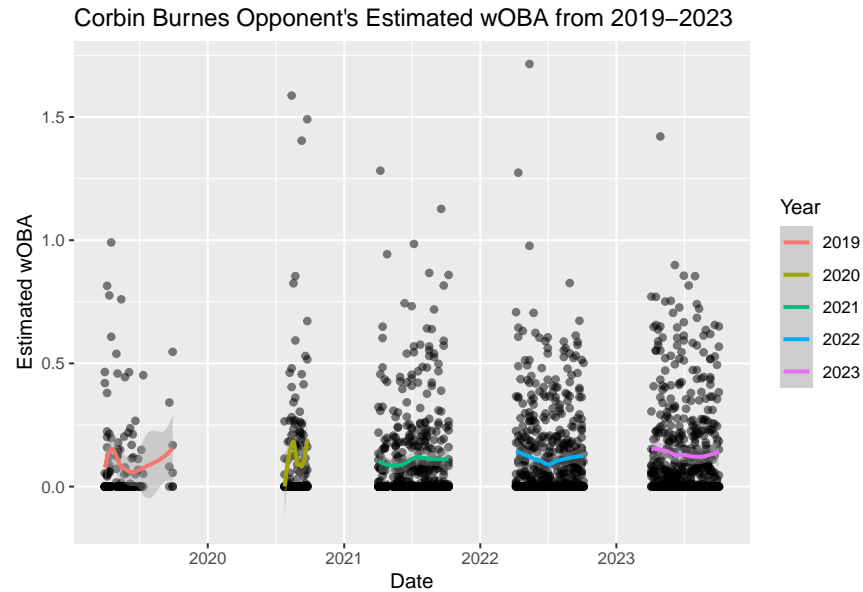
#### Estimated Batting Average:



We can see that Burnes' best pitches from 2019 and 2020 brought him the best results across our five-year interval, but they also brought him the worst results from what we would consider to be good pitches. This speaks to the overall inconsistency Burnes experienced in his career prior to finding a groove in 2021, something he'd carry on in 2022 and beyond.

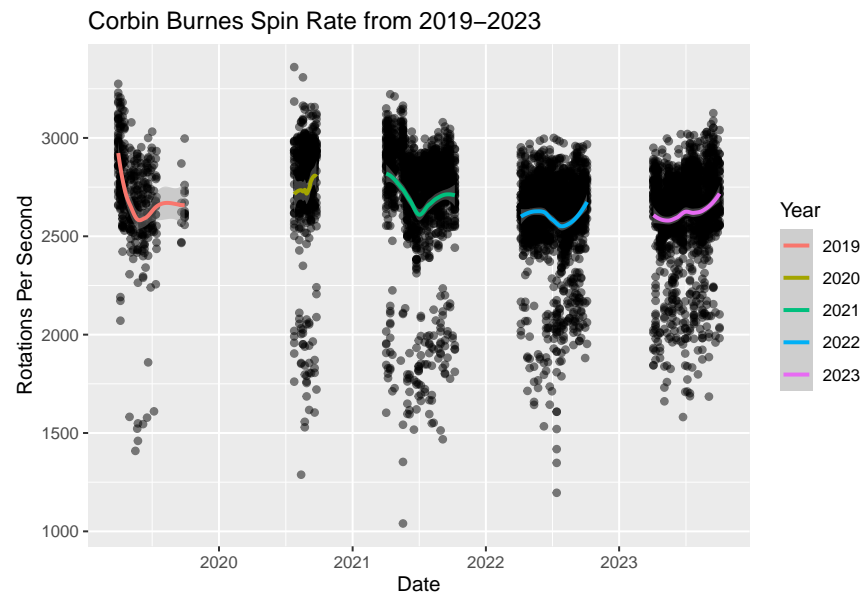
#### Estimated wOBA:





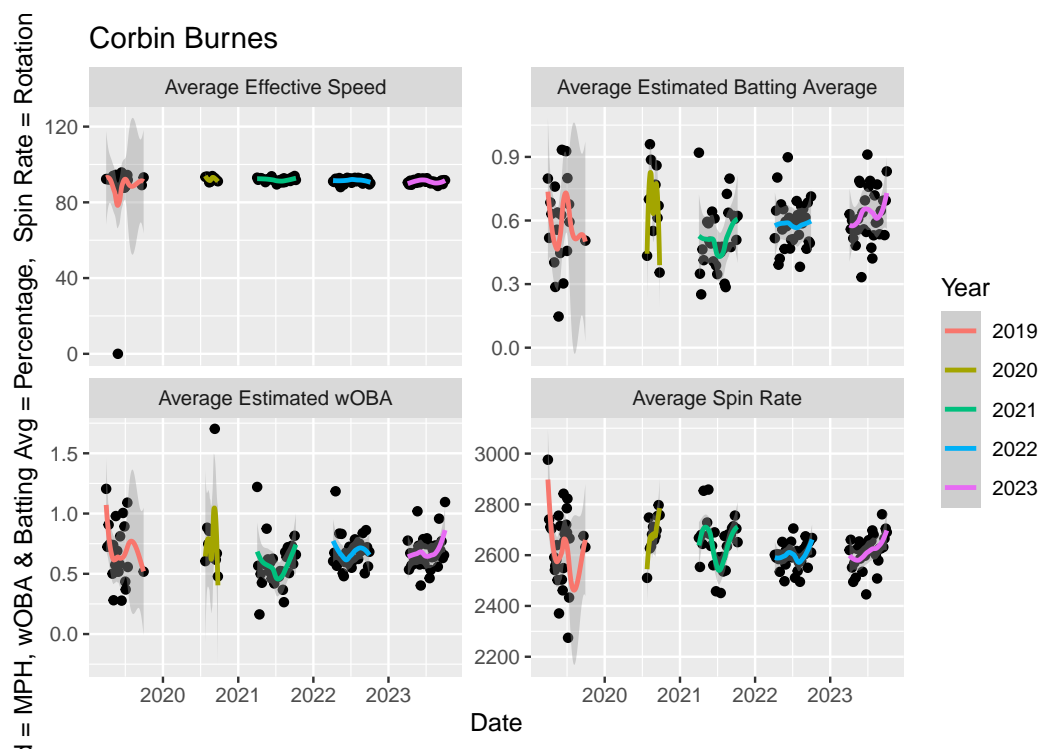
We can once again point out that Burnes' good pitches resulted in much tighter graphs for 2021-2023 as he figured out his pitching.

### Spin Rate:



What is fascinating to note here is that we are tracking Burnes' pitches across entire seasons. With this, we can see in 2019 how his good pitches start as being ones with high spin rates, then by the middle of the season, his best pitches have less spin, possibly indicating how he has success more so with pitches that break less (hint, it's his cutter). We see this trend again in 2021, with tighter graphs in 2022 and 2023.

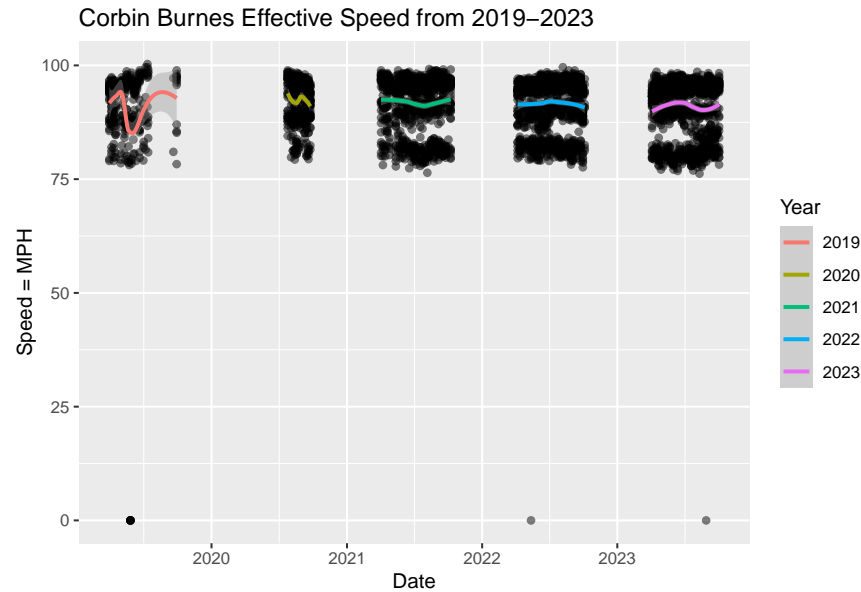
## Graph of Burnes' Bad Pitches from Avg Stats



Looking at the graphs for Burnes' bad pitches from his average metrics, we can see that there is not much of a difference between the average effective speed per start from the good pitches graph and the bad pitches graph, indicating that he succeeded and failed with pitches of all speeds. Where we see a noticeable spike is in the 2019 and 2020 years for average estimated batting average, where pitches deemed as "bad" resulted in estimated batting averages at about .700 and above. These are extremely high averages, worthy of one making poor pitches, so it comes as no surprise that our function for pitch quality had these at negative run expectancies. We see the spikes and variance once again for the 2019 season when it comes to wOBA, and once again more clean and tighter variances for 2022 and 2023 in particular, the sign of a pitcher that has gotten increasingly more comfortable with his repertoire. What we can note from the spin rate graph is how Burnes got poor results from bad pitches of all spin types, especially in 2019. This tells us that Burnes struggled with basically all of his pitches in 2019 due to a more varied pitch mix. The rest of the seasons, especially 2021-2023 tell us that Burnes made bad pitches with a more selective pitch mix, likely his cutter. Comparing this to the good pitches graph, we can infer that when Burnes does not spin his cutting fastball enough, his pitches have poor run expectancy, and when he gets a good amount of spin, his pitches' run expectancy is over zero.

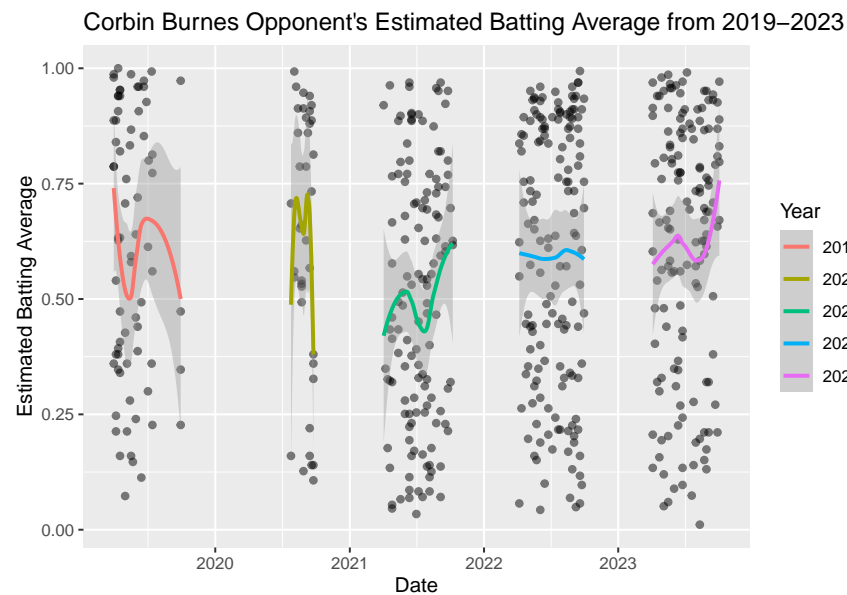
## Graphs of Burnes' Bad Pitches from Raw Stats

### Effective Speed:



The graph for effective speed indicates that Corbin Burnes' effective speed of bad pitches varied significantly more in 2019 than it did in the following 4 years. At the beginning and end of the 2019 season, Burnes consistently threw bad pitches at a high speed. In the middle of the 2019 season, Burnes' effective speed of bad pitches dropped to approximately 85 MPH. This drop in effective speed might imply that Burnes was struggling to adjust to the MLB early in his career. As a result, he attempted to change strategies and throw different types of pitches at a lower speed. Unfortunately, this adjustment in strategy still resulted in bad pitches. In the 2020 season, Burnes only played approximately half the games as normal due to Covid-19. Throughout 2021 to 2023, the effective speed of Burnes's bad pitches illustrated relatively fast speed with minor variations.

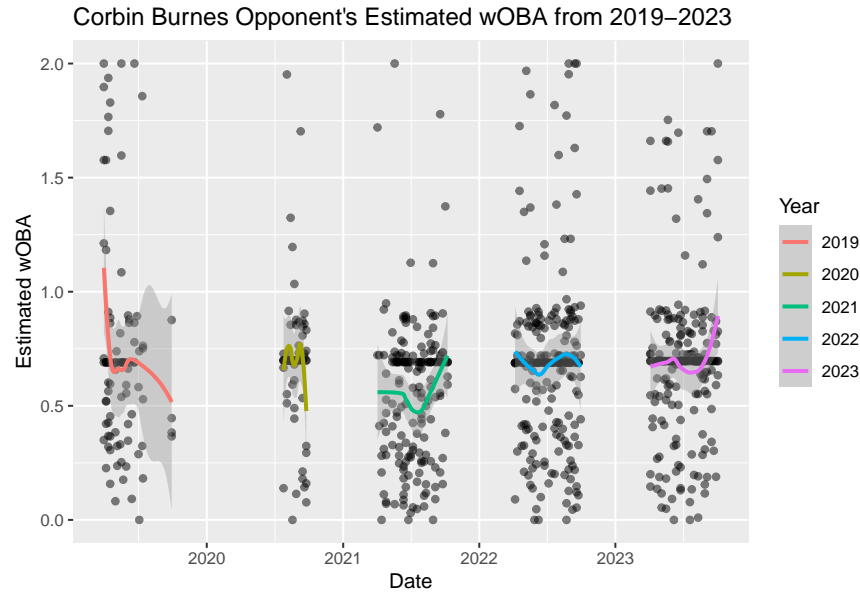
#### Estimated Batting Average:



Corbin Burnes' Opponent's Estimated Batting Average peaked at the beginning of the 2019 season and at the end of the 2023 season. Considering that low estimated batting average indicates good performance from the pitcher, these were times when Burnes was struggling. At the start of the 2021 season, Corbin Burnes' estimated batting average for bad pitches was significantly lower than previous years, indicating that he was

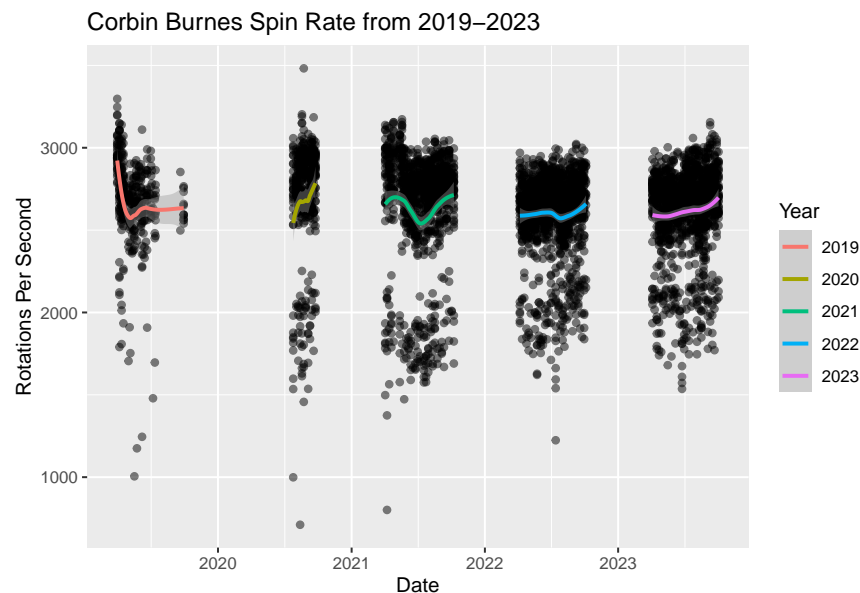
throwing his best pitches. For the rest of the 2021 season, his batting average fluctuated and slowly but steadily increased. By the end of the 2021 season, Burnes' batting average for bad pitches had increased to 0.62, indicating that his opponents were performing better against him. This batting average remained relatively consistent until the end of the 2023 season, where his batting average significantly increased to almost 0.8, indicating that Burnes was performing extremely poorly.

#### wOBA:



Similar to EBA, the Weighted Opponent's Batting Average for Corbin Burnes was significantly higher at the beginning of 2019 and at the end of 2023. Considering that this batting average estimate is weighted, this measurement is more accurate than EBA. We can see a significant drop in wOBA for Burnes' bad pitches in the middle of 2021 and at the end of 2020. This graph further supports our analysis of the EBA graph.

#### Spin Rate:



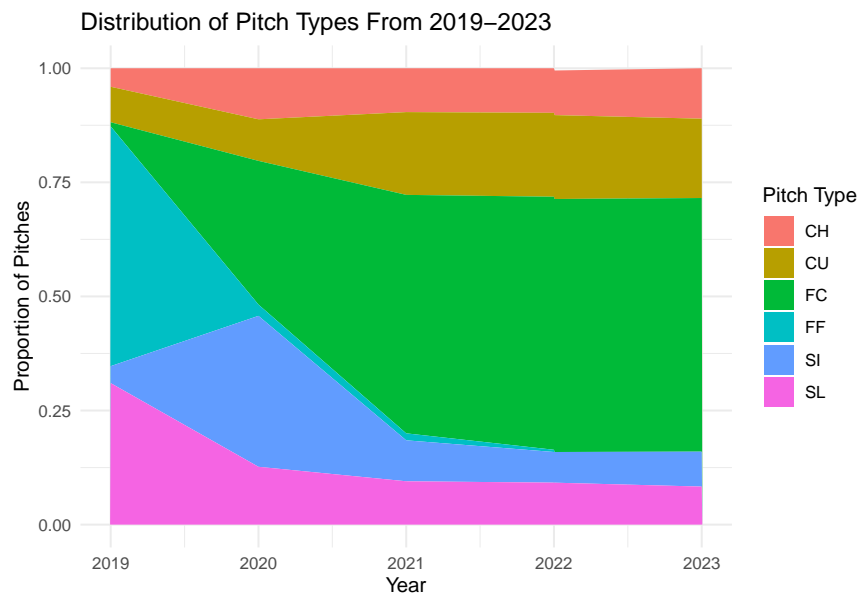
The Spin Rate of Corbin Burnes' bad pitches significantly dropped after the beginning of the 2019 season. In 2020, Burnes started the season with a slower spin rate and consistently ramped up throughout the whole

season. During the 2021 season, he started and finished the season with a similar spin rate, but used less spin in the middle of the season. For both 2022 and 2023, Corbin Burnes started the season with a relatively low spin rate and steadily increased throughout the season.

## Observing Burnes' Pitch Type Distribution

An integral part to a pitcher's success is the various pitches they have in their arsenal. Burnes specifically provides a great example of how a pitcher can greatly improve after honing a specific pitch. As we have displayed earlier, Burnes has greatly improved from his second season (2019) – largely because he was able to develop a Cutter pitch. We'll explore how this distribution changed over time.

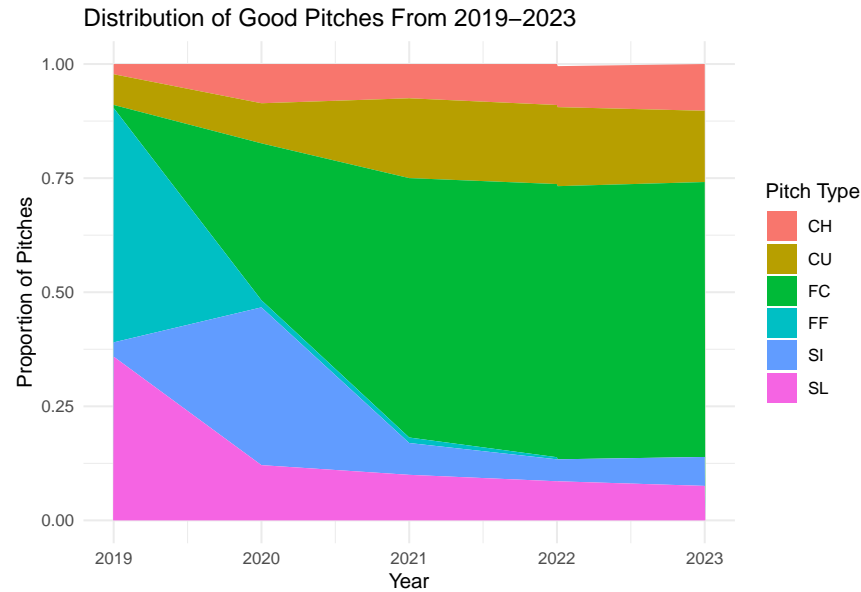
```
## 'summarise()' has grouped output by 'year'. You can override using the
## '.groups' argument.
```



Over this five year span, we see a drastic change in Burnes gameplan. While his fastball initially had the lions share of his pitches, he quickly transitioned towards developing a fastball cutter. By 2021, he rarely threw any fastballs. He also noticeably relies less on his slider. In 2023, his pitch distribution numerically was 55.5% Cutters, 17.4% Curveballs, 11.1% Changeups, 8.3% Sliders, 7.7% Sinkers, and 0 fastballs.

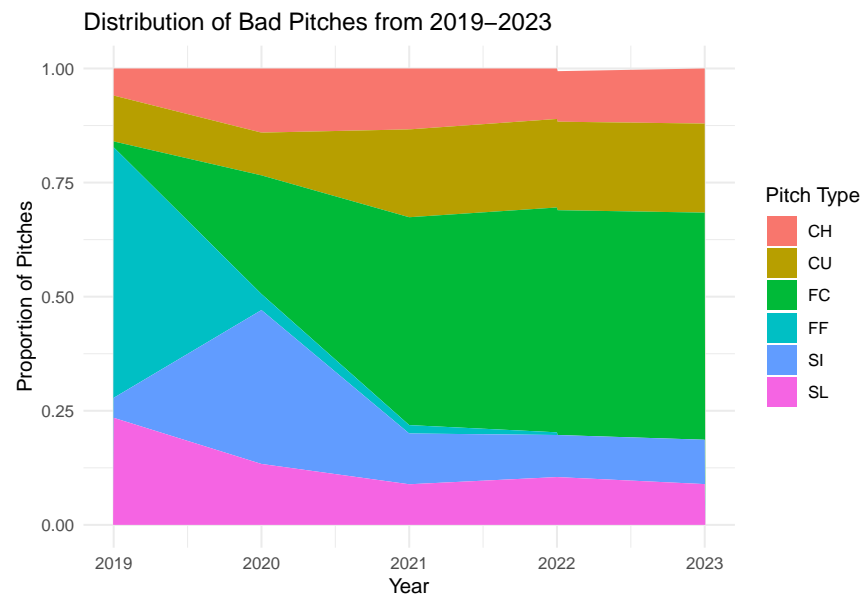
From here, we can start to look which types of pitches have tended to yield negative delta run expectancy (good pitches) and a positive delta run expectancy (bad pitches).

```
## 'summarise()' has grouped output by 'year'. You can override using the
## '.groups' argument.
```



Above, we can see that the distribution of pitching types for good pitches resembles the overall distribution very closely. This can be attributed to the fact that as shown above, the majority of his pitches are considered “good”. The largest notable change is that while making up 55% of his overall pitches, cutters make up **60% of his good pitches in 2023**.

## ‘summarise()’ has grouped output by ‘year’. You can override using the  
## ‘.groups’ argument.



There are some noticeable changes with the bad pitch distribution from the overall pitch distribution. While Cutters constitute 55% of his overall pitches and 60% of his good pitches, they make up only 50% of his bad pitches. Looking solely at the raw numbers, in 2023, if Burnes were to throw a cutter, it is **1.56x** more likely to decrease the opposing batters chances of scoring than it would increase their chances of scoring.

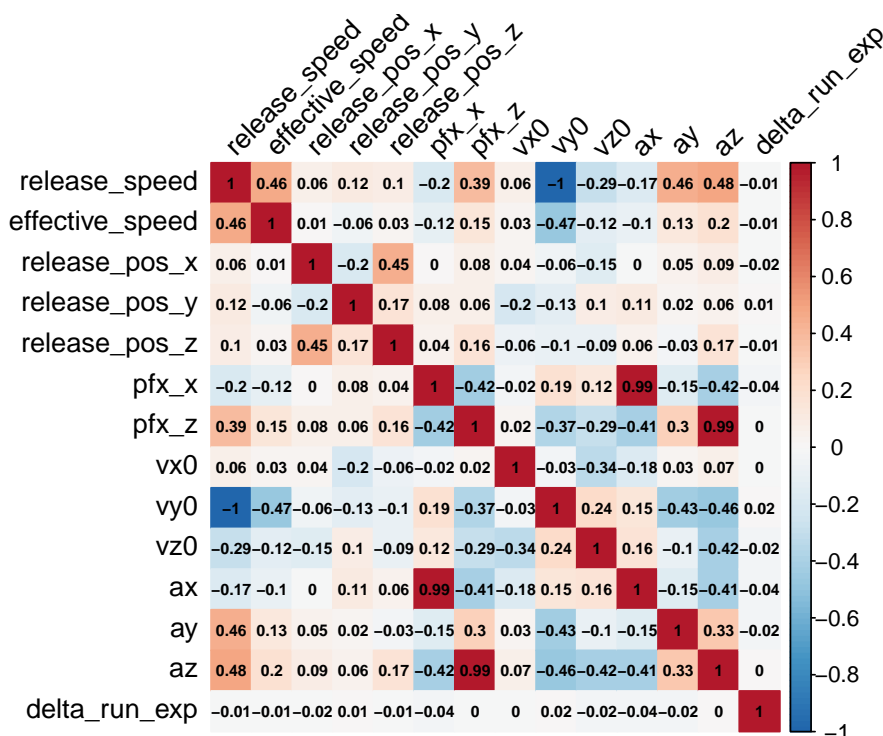
## Correlation Heatmaps by Pitch

The following graphs are heatmaps, designed to show how correlated the variables are to each other. We define these variables as such:

release\_speed, speed of pitch (mph) when released effective\_speed, speed of pitch (mph) when crossing plate  
 release\_pos\_x, x-coordinate of release point of pitch release\_pos\_y, y-coordinate of release point of pitch  
 release\_pos\_z, z-coordinate of release point of pitch pfx\_x, horizontal movement of pitch pfx\_z, vertical  
 movement of pitch vx0, x-coordinate of pitch velocity vy0, y-coordinate of pitch velocity vz0, z-coordinate  
 of pitch velocity ax, x-coordinate of pitch acceleration ay, y-coordinate of pitch acceleration az, z-coordinate  
 of pitch acceleration delta\_run\_exp, expected chance of scoring

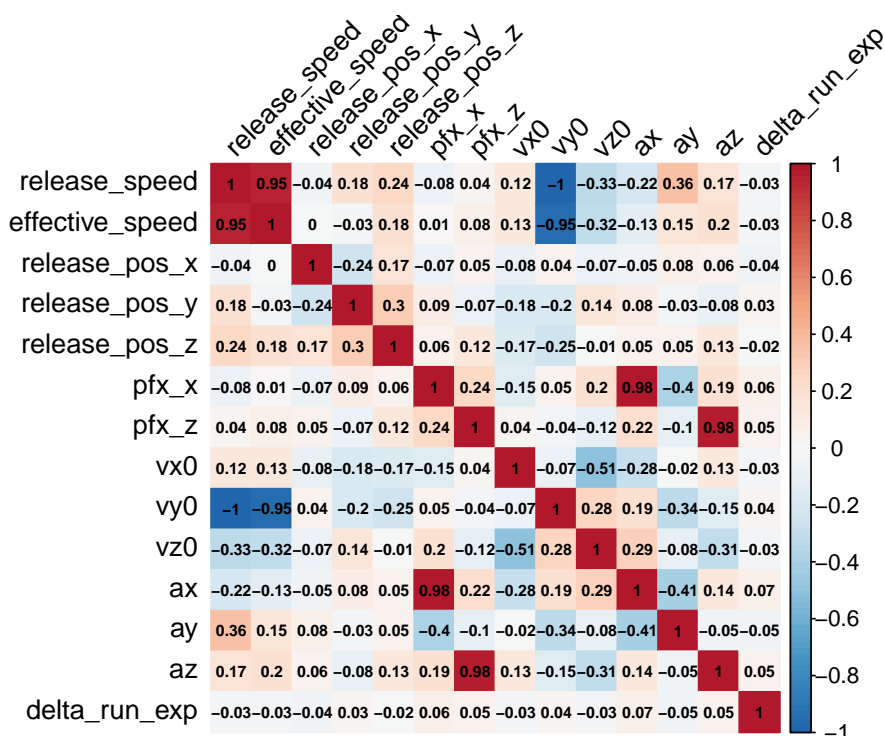
On the heatmap, the boxes shaded red depict a positive correlation between the variables, and the boxes shaded blue depict a negative correlation between the variables. We create a heatmap for each pitch Burnes throws, and we take a look at each variable's relationship with our overall metric of run expectancy. We will look at Burnes five primary pitches. Fastballs are noticeably absent as they constitute 0-1% of his pitches after 2019.

**Cutters:**



We can pinpoint the hottest and coldest zones in this heatmap for Burnes' cutters. We see that for cutters, release speed and the y-coordinate for pitch velocity have a negative correlation in terms of change in run expectancy, meaning that hitters tend to do worse in terms of this relationship. Of note, we can also pinpoint the negatively correlated relationship of effective speed and the y-coordinate for pitch velocity, and the negatively correlated relationship between the z-coordinate of pitch acceleration and the y-coordinate of pitch velocity. These two relationships result in a change in run expectancy of -0.47 and -0.46, respectively. What we can also note for this heatmap is that the cutter is known as Burnes' most effective pitch and also a variation of a fastball, which can explain some of these relationships. In terms of positively correlated relationships, we see that there is one between horizontal movement of pitch and the x-coordinate of pitch acceleration and one between vertical movement of pitch and z-coordinate of pitch position. Both relationships have a +0.99 change in run expectancy, indicating that these relationships between variables cause Burnes to allow runs and struggle. We can note that these variables are linked through axes, with the relationships following the horizontal or vertical axes and matching them.

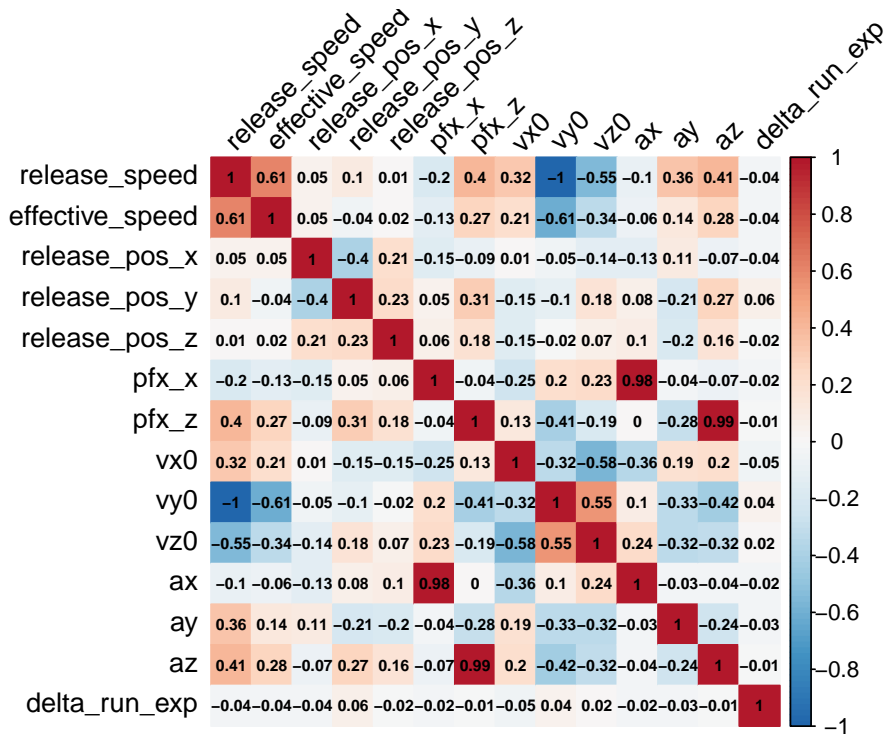
## Changeup:



Moving on to the correlation heatmap for Burnes' changeups, we see notable positively correlated relationships between release speed and effective speed, and horizontal movement of pitch and the x-coordinate of pitch acceleration. These two relationships have change in run expectancies of almost +1, indicating that these relationships do not bode well for Burnes' changeups. On the other hand, we see that there is a negatively correlated relationship between release speed and the y-coordinate of pitch velocity and between effective speed and the y-coordinate of pitch velocity. Noting that the changeup pitch is reliant on tricking the hitter on its speed, it makes sense that these two variable relationships can cause hitters to struggle against Burnes' changeups.

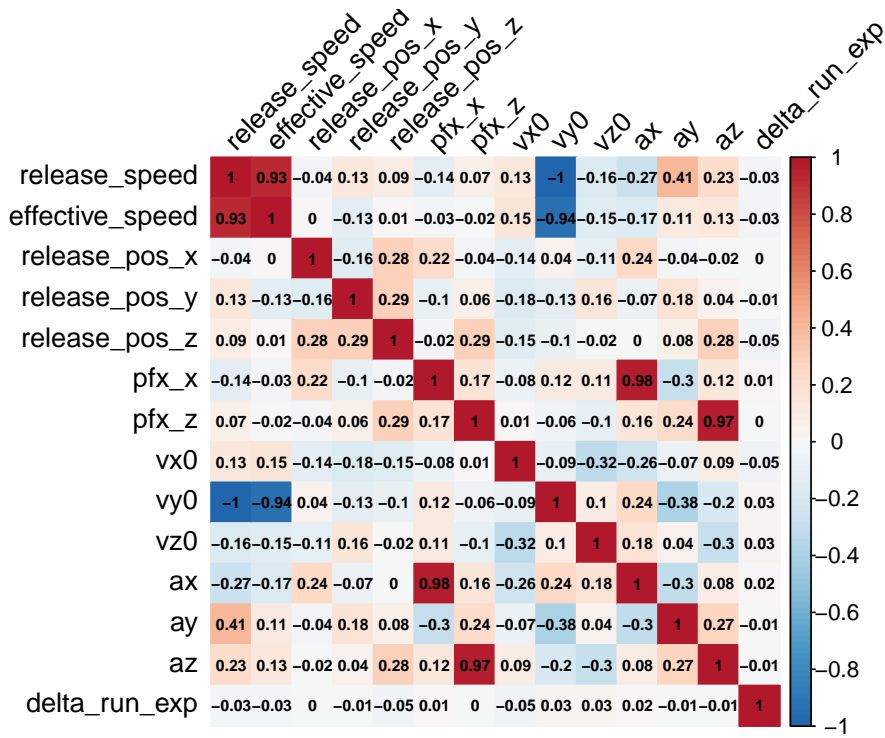
## Curveballs:





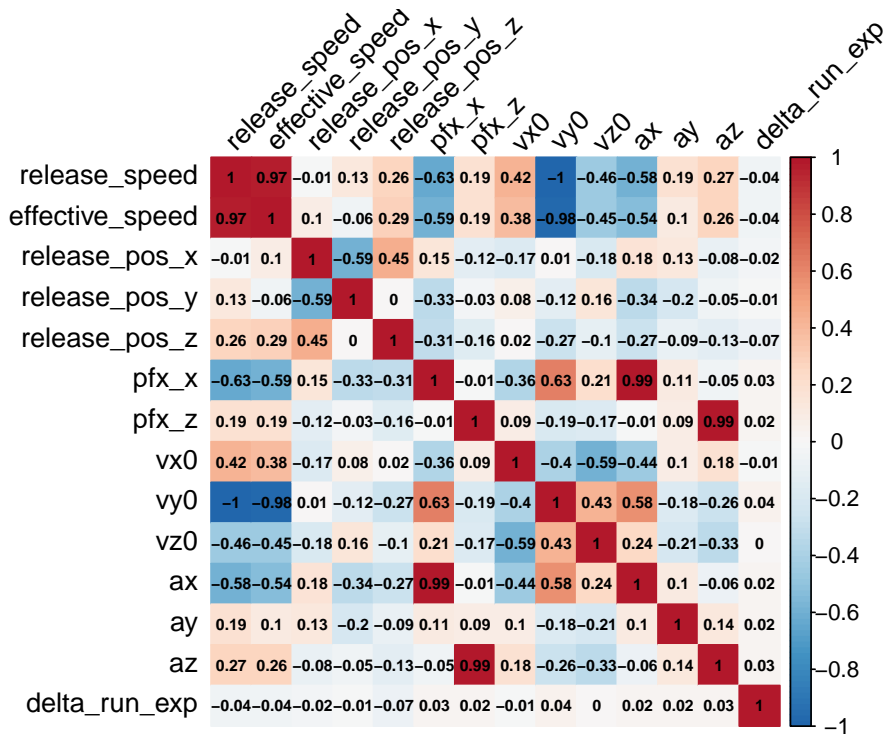
For Burnes' curveballs, we see that there are positively correlated relationships between horizontal movement of pitch and the x-coordinate of pitch acceleration and vertical movement of pitch and the z-coordinate of pitch acceleration of nearly +1 change in run expectancy, indicating that these variables can lead hitters to have success against Burnes' curveballs. Analyzing this success, it would make sense that these variables are important to the change in run expectancy considering that the curveball pitch is reliant on its break and location. Curveballs usually tend to have a lot of vertical or horizontal movement, and it appears as though hitters have success against Burnes in this regard. Connecting this to our prior "bad" pitches graph, these curveballs may be the pitches with high spin rate that lie in that graph, especially since curveballs have a lot of spin on them. Burnes sees success in the relationships between the y-coordinate of pitch velocity and release speed, between effective speed and the y-coordinate of pitch velocity, and between the z-coordinate of pitch velocity and release speed, at -1, -0.61, and -0.55 changes in run expectancy, respectively. Due to curveballs being slower pitches, it would make sense that Burnes sees success from the speed aspect of these pitches, since they may catch hitters off-guard.

**Sinkers:**



Moving on to Burnes' sinker pitches, we note that Burnes struggles in the regard of the positively correlated relationship between the horizontal movement of the pitch and the x-coordinate of pitch acceleration and with the positively correlated relationship between the vertical movement of the pitch and the z-coordinate of pitch acceleration, as well as the relationship between release speed and effective speed. Explaining these relationships, it could be that Burnes does not have enough "sink" or break, on his sinkers, causing hitters to have success against them, as well as potentially lacking enough velocity to be successful. On the other hand, we see that Burnes has success in terms of having a negative change in run expectancy with the negative relationships of the y-coordinate of pitch velocity with effective speed and release speed, once again implying that it may matter where Burnes locates his sinker in order for him to be successful.

**Sliders:**



For Burnes' heatmap for sliders, there are many "hot" and "cold" zones to examine. Skimming through the graph, we can see that where Burnes is successful in terms of preventing runs is when he masters his speeds with where he locates his slider. These relationships have changes in run expectancies between -0.45 and -1, while the relationships of horizontal movement of pitch with the two speeds have changes in run expectancies of -0.63 and -0.59. For the latter two relationships, this makes a lot of sense considering that the slider is a pitch that must have horizontal break to be effective. We see that Burnes struggled with the relationship between effective speed and release speed, which tracks due to the slider not being important in the area of velocity, moreso having break and movement. We also see that the variables that deal with the break of the pitch (horizontal and vertical) have positive relationships with the variables that deal with the acceleration of the pitch.

## Conclusion

Through this report, we sought to analyze MLB pitcher Corbin Burnes and his pitches. From examining Burnes through the lens of the advanced metrics of effective speed, spin rate, estimated batting average, we got a surface-level look at how he became a completely different pitcher after struggling in the 2019 season. By making a function that deemed all of his pitches either "good" or "bad", we were able to use the aforementioned metrics to see how he changed his pitches and how each pitch in every season fared, filtered by either being a "good" or "bad" pitch based on how it changed the run expectancy. We supplemented the findings from our filtered function by looking at how Burnes' pitch distribution has changed over our five-year span, showing us how he developed a cutter pitch and rode it to success in 2020 and beyond. This information goes along with our findings from analyzing his good and bad pitches, because it gives us implications that we can confirm and connect to what we know. For example, Burnes' "good" pitches in his later years in the interval were ones of lesser spin rate. That tells us that those pitches were likely his cutter and not his breaking pitches, which appeared to be regulars on his "bad" pitches graph. Furthermore, we created heatmaps to display how each variable feature has a relationship with each other, giving us an idea of what variable relationships lead to positive or negative run expectancies for Burnes for each type of pitch he throws. This allows us to analyze each pitch and use the context of the pitch itself to make conclusions on why or why not the pitch is successful for Burnes. Ultimately, we were able to illustrate and explain Corbin Burnes career within our five-year interval through not only our created function, but the graphs using the

metrics and variable features available to us.

## Code Appendix

```
install.packages("baseballr", repos = "http://cran.us.r-project.org")
install.packages("tidyverse", repos = "http://cran.us.r-project.org")
install.packages("reshape2", repos = "http://cran.us.r-project.org")
install.packages("zoo", repos = "http://cran.us.r-project.org")
install.packages("corrplot", repos = "http://cran.us.r-project.org")

library(baseballr)
library(tidyverse)
library(reshape2)
library(zoo)
library(corrplot)
options(warn=-1)
csv_from_url <- function(...){
  data.table::fread(...)
}
#-----
make_baseballr_data <- function(df, type, timestamp){
  out <- df %>%
    tidyr::as_tibble()

  class(out) <- c("baseballr_data", "tbl_df", "tbl", "data.table", "data.frame")
  attr(out, "baseballr_timestamp") <- timestamp
  attr(out, "baseballr_type") <- type
  return(out)
}
# @title
statcast_search <- function(start_date = Sys.Date() - 1, end_date = Sys.Date(),
                             playerid = NULL,
                             player_type = "batter", ...) {
  # Check for other user errors.
  if (start_date <= "2015-03-01") { # March 1, 2015 was the first date of Spring Training.
    message("Some metrics such as Exit Velocity and Batted Ball Events have only been compiled since 2015.")
  }
  if (start_date < "2008-03-25") { # March 25, 2008 was the first date of the 2008 season.
    stop("The data are limited to the 2008 MLB season and after.")
    return(NULL)
  }
  if (start_date == Sys.Date()) {
    message("The data are collected daily at 3 a.m. Some of today's games may not be included.")
  }
  if (start_date > as.Date(end_date)) {
    stop("The start date is later than the end date.")
    return(NULL)
  }

  playerid_var <- ifelse(player_type == "pitcher",
                         "pitchers_lookup%5B%5D", "batters_lookup%5B%5D")

  vars <- tibble::tribble(
    ~var, ~value,
    "all", "true",
    "hfPT", "",
    "hfAB", "",
    "hfBBT", "",
    "hfPR", "",
    "hfZ", "",
    "stadium", "",
    "hfBBL", "",
    "hfNewZones", "",
    "hfGT", "R%7CP0%7CS%7C&hfC",
    "hfSea", paste0(lubridate::year(start_date), "%7C"),
    "hfSit", "",
```

```

    "hfOuts", "",
    "opponent", "",
    "pitcher_throws", "",
    "batter_stands", "",
    "hfSA", "",
    "player_type", player_type,
    "hfInfield", "",
    "team", "",
    "position", "",
    "hfOutfield", "",
    "hfRO", "",
    "home_road", "",
    playerid_var, ifelse(is.null(playerid), "", as.character(playerid)),
    "game_date_gt", as.character(start_date),
    "game_date_lt", as.character(end_date),
    "hfFlag", "",
    "hfPull", "",
    "metric_1", "",
    "hfInn", "",
    "min_pitches", "0",
    "min_results", "0",
    "group_by", "name",
    "sort_col", "pitches",
    "player_event_sort", "h_launch_speed",
    "sort_order", "desc",
    "min_abs", "0",
    "type", "details") %>%
dplyr::mutate(pairs = paste0(.data$var, "=", .data$value))

if (is.null(playerid)) {
  # message("No playerid specified. Collecting data for all batters/pitchers.")
  vars <- vars %>%
    dplyr::filter(!grepl("lookup", .data$var))
}

url_vars <- paste0(vars$pairs, collapse = "&")
url <- paste0("https://baseballsavant.mlb.com/statcast_search/csv?", url_vars)
# message(url)

# Do a try/catch to show errors that the user may encounter while downloading.
tryCatch(
  {
    suppressMessages(
      suppressWarnings(
        payload <- csv_from_url(url, encoding = "UTF-8")
      )
    )
  },
  error = function(cond) {
    message(cond)
    stop("No payload acquired")
  },
  # this will never run??
  warning = function(cond) {
    message(cond)
  }
)
# returns 0 rows on failure but > 1 columns
if (nrow(payload) > 1) {
  names(payload) <- c("pitch_type", "game_date", "release_speed", "release_pos_x",
    "release_pos_z", "player_name", "batter", "pitcher", "events",
    "description", "spin_dir", "spin_rate_deprecated", "break_angle_deprecated",
    "break_length_deprecated", "zone", "des", "game_type", "stand",
    "p_throws", "home_team", "away_team", "type", "hit_location",
    "bb_type", "balls", "strikes", "game_year", "pfx_x", "pfx_z",
    "plate_x", "plate_z", "on_3b", "on_2b", "on_1b", "outs_when_up",

```

```

        "inning", "inning_topbot", "hc_x", "hc_y", "tfs_deprecated",
        "tfs_zulu_deprecated", "fielder_2", "umpire", "sv_id", "vx0",
        "vy0", "vz0", "ax", "ay", "az", "sz_top", "sz_bot", "hit_distance_sc",
        "launch_speed", "launch_angle", "effective_speed", "release_spin_rate",
        "release_extension", "game_pk", "pitcher_1", "fielder_2_1",
        "fielder_3", "fielder_4", "fielder_5", "fielder_6", "fielder_7",
        "fielder_8", "fielder_9", "release_pos_y", "estimated_ba_using_speedangle",
        "estimated_woba_using_speedangle", "woba_value", "woba_denom",
        "babip_value", "iso_value", "launch_speed_angle", "at_bat_number",
        "pitch_number", "pitch_name", "home_score", "away_score", "bat_score",
        "fld_score", "post_away_score", "post_home_score", "post_bat_score",
        "post_fld_score", "if_fielding_alignment", "of_fielding_alignment",
        "spin_axis", "delta_home_win_exp", "delta_run_exp", "bat_speed", "swing_length")
payload <- process_statcast_payload(payload) %>%
  make_baseballr_data("MLB Baseball Savant Statcast Search data from baseballsavant.mlb.com", Sys.time())
return(payload)
} else {
  warning("No valid data found")

# (somewhere within the statcast_search function before the payload is searched for)
colos <- c("pitch_type", "game_date",
  "release_speed", "release_pos_x", "release_pos_z",
  "player_name", "batter", "pitcher",
  "events", "description", "spin_dir",
  "spin_rate_deprecated", "break_angle_deprecated",
  "break_length_deprecated", "zone", "des",
  "game_type", "stand", "p_throws",
  "home_team", "away_team", "type",
  "hit_location", "bb_type", "balls",
  "strikes", "game_year", "pfx_x",
  "pfx_z", "plate_x", "plate_z",
  "on_3b", "on_2b", "on_1b", "outs_when_up",
  "inning", "inning_topbot", "hc_x",
  "hc_y", "tfs_deprecated", "tfs_zulu_deprecated",
  "fielder_2", "umpire", "sv_id",
  "vx0", "vy0", "vz0", "ax",
  "ay", "az", "sz_top", "sz_bot",
  "hit_distance_sc", "launch_speed", "launch_angle",
  "effective_speed", "release_spin_rate",
  "release_extension", "game_pk", "pitcher_1",
  "fielder_2_1", "fielder_3", "fielder_4",
  "fielder_5", "fielder_6", "fielder_7",
  "fielder_8", "fielder_9", "release_pos_y",
  "estimated_ba_using_speedangle", "estimated_woba_using_speedangle",
  "woba_value", "woba_denom", "babip_value",
  "iso_value", "launch_speed_angle", "at_bat_number",
  "pitch_number", "pitch_name", "home_score",
  "away_score", "bat_score", "fld_score",
  "post_away_score", "post_home_score",
  "post_bat_score", "post_fld_score", "if_fielding_alignment",
  "of_fielding_alignment", "spin_axis",
  "delta_home_win_exp", "delta_run_exp")
colNumber <- ncol(payload)
if(length(colos) != colNumber){
  newCols <- paste("newStat", 1:(length(colos) - colNumber))
  colos <- c(colos, newCols)
  message("New stats detected! baseballr will be updated soon to properly identify these stats")
}

# payload is acquired somewhere in here
# when the payload columns need to be named:
names(payload) <- colos

payload <- payload %>%
  make_baseballr_data("MLB Baseball Savant Statcast Search data from baseballsavant.mlb.com", Sys.time())

```

```

    return(payload)
  }
}

statcast_search.default <- function(start_date = Sys.Date() - 1, end_date = Sys.Date(),
                                     playerid = NULL, player_type = "batter", ...) {

  message(paste0(start_date, " is not a date. Attempting to coerce..."))
  start_Date <- as.Date(start_date)

  tryCatch(
    {
      end_Date <- as.Date(end_date)
    },
    warning = function(cond) {
      message(paste0(end_date, " was not coercible into a date. Using today. "))
      end_Date <- Sys.Date()
      message("Original warning message:")
      message(cond)
    }
  )

  statcast_search(start_Date, end_Date,
                  playerid, player_type, ...)
}

statcast_search_batters <- function(start_date, end_date, batterid = NULL, ...) {
  statcast_search(start_date, end_date, playerid = batterid,
                  player_type = "batter", ...)
}

statcast_search_pitchers <- function(start_date, end_date, pitcherid = NULL, ...) {
  statcast_search(start_date, end_date, playerid = pitcherid,
                  player_type = "pitcher", ...)
}

burnes_19 <- statcast_search(start_date = "2019-03-31", end_date = "2019-10-31", playerid = 669203, player_type = 'pitcher')
burnes_20 <- statcast_search(start_date = "2020-07-23", end_date = "2020-10-31", playerid = 669203, player_type = 'pitcher')
burnes_21 <- statcast_search(start_date = "2021-03-31", end_date = "2021-10-31", playerid = 669203, player_type = 'pitcher')
burnes_22 <- statcast_search(start_date = "2022-03-31", end_date = "2022-10-31", playerid = 669203, player_type = 'pitcher')
burnes_23 <- statcast_search(start_date = "2023-03-31", end_date = "2023-10-31", playerid = 669203, player_type = 'pitcher')

burnes_19 = burnes_19 %>% arrange(game_date)
burnes_20 = burnes_20 %>% arrange(game_date)
burnes_21 = burnes_21 %>% arrange(game_date)
burnes_22 = burnes_22 %>% arrange(game_date)
burnes_23 = burnes_23 %>% arrange(game_date)
burnes_total <- bind_rows(burnes_19, burnes_20, burnes_21, burnes_22, burnes_23) %>%
  mutate(Year = as.factor(substr(game_date, 1, 4)))

head(burnes_total)
burnes <- burnes_total %>%
  group_by(game_date) %>%
  summarise(
    `Average Effective Speed` = mean(effective_speed, na.rm = TRUE),
    `Average Estimated Batting Average` = mean(estimated_ba_using_speedangle, na.rm = TRUE),
    `Average Estimated wOBA` = mean(estimated_woba_using_speedangle, na.rm = T),
    `Average Spin Rate` = mean(release_spin_rate, na.rm = T) %>%
  ungroup() %>%
  melt(id=c("game_date")) %>%
  mutate(Year = as.factor(substr(game_date, 1, 4)))

burnes_all_table <- burnes %>%
  filter(value != "NaN")

head(burnes_all_table)

```

```

burnes_avg_yr <- burnes_total %>%
  group_by(Year) %>%
  summarise(
    Estimated_Batting_Average = round(mean(estimated_ba_using_speedangle, na.rm = TRUE),3),
    Effective_Speed = round(mean(effective_speed, na.rm = TRUE),3),
    Estimated_wOBA = round(mean(estimated_woba_using_speedangle, na.rm = TRUE), 3),
    Spin_Rate = round(mean(release_spin_rate, na.rm = TRUE), 3))
burnes_avg_yr
suppressWarnings(burnes %>%
  ggplot(aes(game_date, value)) +
  geom_point() +
  stat_smooth(aes(group = Year, color = Year),formula = y~x, method='loess') +
  facet_wrap(~variable, scales = "free_y") +
  ggtitle("Corbin Burnes") +
  ylab("Effective Speed = MPH, wOBA & Batting Avg = Percentage, Spin Rate = Rotations per Second") +
  xlab("Date"))
suppressWarnings(burnes_total %>%
  ggplot(aes(game_date, effective_speed)) +
  geom_point(alpha = 0.5) +
  stat_smooth(aes(group = Year, color = Year),formula = y~x, method='loess')+
  ggtitle("Corbin Burnes Effective Speed from 2019-2023") +
  ylab("Speed = MPH") +
  xlab("Date"))
suppressWarnings(burnes_total %>%
  ggplot(aes(game_date, estimated_ba_using_speedangle)) +
  geom_point(alpha = 0.5) +
  stat_smooth(aes(group = Year, color = Year),formula = y~x, method='loess')+
  ggtitle("Corbin Burnes Estimated Batting Average from 2019-2023") +
  ylab("Batting Average") +
  xlab("Date"))
suppressWarnings(burnes_total %>%
  ggplot(aes(game_date, estimated_woba_using_speedangle)) +
  geom_point(alpha = 0.5) +
  stat_smooth(aes(group = Year, color = Year),formula = y~x, method='loess')+
  ggtitle("Corbin Burnes wOBA from 2019-2023") +
  ylab("wOBA") +
  xlab("Date"))
suppressWarnings(burnes_total %>%
  ggplot(aes(game_date, release_spin_rate)) +
  geom_point(alpha = 0.5) +
  stat_smooth(aes(group = Year, color = Year),formula = y~x, method='loess')+
  ggtitle("Corbin Burnes Spin Rate from 2019-2023") +
  ylab("Rotations Per Second") +
  xlab("Date"))
burnes_bad_pitch <- bind_rows(burnes_19, burnes_20,burnes_21, burnes_22,burnes_23) %>%
  mutate(Year = as.factor(substr(game_date,1,4))) %>%
  filter(delta_run_exp > 0)
burnes_good_pitch <- bind_rows(burnes_19, burnes_20,burnes_21, burnes_22,burnes_23) %>%
  mutate(Year = as.factor(substr(game_date,1,4))) %>%
  filter(delta_run_exp < 0)
burnes_good_grpd <- burnes_good_pitch %>%
  group_by(game_date) %>%
  summarise(
    `Average Effective Speed` = mean(effective_speed, na.rm = TRUE),
    `Average Estimated Batting Average` = mean(estimated_ba_using_speedangle, na.rm = TRUE),
    `Average Estimated wOBA` = mean(estimated_woba_using_speedangle, na.rm = T),
    `Average Spin Rate` = mean(release_spin_rate, na.rm = T)) %>%
  ungroup() %>%
  melt(id=c("game_date")) %>%
  mutate(Year = as.factor(substr(game_date,1,4)))
burnes_good_table <- burnes_good_grpd %>%
  filter(value != "NaN")

```



```

burnes_bad_grpd <- burnes_bad_pitch %>%
group_by(game_date) %>%
summarise(
  `Average Effective Speed` = mean(effective_speed, na.rm = TRUE),
  `Average Estimated Batting Average` = mean(estimated_ba_using_speedangle, na.rm = TRUE),
  `Average Estimated wOBA` = mean(estimated_woba_using_speedangle, na.rm = T),
  `Average Spin Rate` = mean(release_spin_rate, na.rm = T)) %>%
ungroup() %>%
melt(id=c("game_date")) %>%
mutate(Year = as.factor(substr(game_date,1,4)))

burnes_bad_table <- burnes_bad_grpd %>%
  filter(value != "NaN")
burnes_good_avg_yr <- burnes_good_pitch %>%
  group_by(Year) %>%
  summarise(
    Estimated_Batting_Average = round(mean(estimated_ba_using_speedangle, na.rm = TRUE),3),
    Effective_Speed = round(mean(effective_speed, na.rm = TRUE),3),
    Estimated_wOBA = round(mean(estimated_woba_using_speedangle, na.rm = TRUE), 3),
    Spin_Rate = round(mean(release_spin_rate, na.rm = TRUE), 3))
burnes_good_avg_yr

burnes_bad_avg_yr <- burnes_bad_pitch %>%
  group_by(Year) %>%
  summarise(
    Estimated_Batting_Average = round(mean(estimated_ba_using_speedangle, na.rm = TRUE),3),
    Effective_Speed = round(mean(effective_speed, na.rm = TRUE),3),
    Estimated_wOBA = round(mean(estimated_woba_using_speedangle, na.rm = TRUE), 3),
    Spin_Rate = round(mean(release_spin_rate, na.rm = TRUE), 3))
burnes_bad_avg_yr

suppressWarnings(burnes_good_grpd%>%
  ggplot(aes(game_date, value)) +
  geom_point() +
  stat_smooth(aes(group = Year, color = Year),formula = y~x, method='loess') +
  facet_wrap(~variable, scales = "free_y") +
  ggtitle("Corbin Burnes") +
  ylab("Effective Speed = MPH, wOBA & Batting Avg = Percentage, Spin Rate = Rotations per Second") +
  xlab("Date"))

suppressWarnings(burnes_good_pitch %>%
  ggplot(aes(game_date, effective_speed)) +
  geom_point(alpha = 0.5) +
  stat_smooth(aes(group = Year, color = Year),formula = y~x, method='loess')+
  ggtitle("Corbin Burnes Effective Speed from 2019-2023") +
  ylab("Speed = MPH") +
  xlab("Date"))

suppressWarnings(burnes_good_pitch %>%
  ggplot(aes(game_date, estimated_ba_using_speedangle)) +
  geom_point(alpha = 0.5) +
  stat_smooth(aes(group = Year, color = Year),formula = y~x, method='loess')+
  ggtitle("Corbin Burnes Estimated Opponent Batting Average from 2019-2023") +
  ylab("Estimated Batting Average") +
  xlab("Date"))

suppressWarnings(burnes_good_pitch %>%
  ggplot(aes(game_date, estimated_woba_using_speedangle)) +
  geom_point(alpha = 0.5) +
  stat_smooth(aes(group = Year, color = Year),formula = y~x, method='loess')+
  ggtitle("Corbin Burnes Opponent's Estimated wOBA from 2019-2023") +
  ylab("Estimated wOBA") +
  xlab("Date"))

suppressWarnings(burnes_good_pitch %>%
  ggplot(aes(game_date, release_spin_rate)) +
  geom_point(alpha = 0.5) +
  stat_smooth(aes(group = Year, color = Year),formula = y~x, method='loess')+
  ggtitle("Corbin Burnes Spin Rate from 2019-2023") +
  ylab("Rotations Per Second") +
  xlab("Date"))

```

```

suppressWarnings(burnes_bad_grpd%>%
  ggplot(aes(game_date, value)) +
  geom_point() +
  stat_smooth(aes(group = Year, color = Year), formula = y~x, method='loess') +
  facet_wrap(~variable, scales = "free_y") +
  ggtitle("Corbin Burnes") +
  ylab("Effective Speed = MPH, wOBA & Batting Avg = Percentage, Spin Rate = Rotations per Second") +
  xlab("Date"))
suppressWarnings(burnes_bad_pitch %>%
  ggplot(aes(game_date, effective_speed)) +
  geom_point(alpha = 0.5) +
  stat_smooth(aes(group = Year, color = Year), formula = y~x, method='loess')+
  ggtitle("Corbin Burnes Effective Speed from 2019-2023") +
  ylab("Speed = MPH") +
  xlab("Date"))
suppressWarnings(burnes_bad_pitch %>%
  ggplot(aes(game_date, estimated_ba_using_speedangle)) +
  geom_point(alpha = 0.5) +
  stat_smooth(aes(group = Year, color = Year), formula = y~x, method='loess')+
  ggtitle("Corbin Burnes Opponent's Estimated Batting Average from 2019-2023") +
  ylab("Estimated Batting Average") +
  xlab("Date"))
suppressWarnings(burnes_bad_pitch %>%
  ggplot(aes(game_date, estimated_woba_using_speedangle)) +
  geom_point(alpha = 0.5) +
  stat_smooth(aes(group = Year, color = Year), formula = y~x, method='loess')+
  ggtitle("Corbin Burnes Opponent's Estimated wOBA from 2019-2023") +
  ylab("Estimated wOBA") +
  xlab("Date"))
suppressWarnings(burnes_bad_pitch %>%
  ggplot(aes(game_date, release_spin_rate)) +
  geom_point(alpha = 0.5) +
  stat_smooth(aes(group = Year, color = Year), formula = y~x, method='loess')+
  ggtitle("Corbin Burnes Spin Rate from 2019-2023") +
  ylab("Rotations Per Second") +
  xlab("Date"))
burnes_history = bind_rows(burnes_19, burnes_20, burnes_21, burnes_22, burnes_23)

burnes_history$year <- as.numeric(format(as.Date(burnes_history$game_date, format = "%Y-%m-%d"), "%Y"))
burnes_history = burnes_history %>% filter(pitch_type != "")

pitch_summary <- burnes_history %>%
  group_by(year, pitch_type) %>%
  summarize(count = n()) %>%
  ungroup()

pitch_summary <- pitch_summary %>%
  group_by(year) %>%
  mutate(total_pitches = sum(count)) %>%
  ungroup() %>%
  mutate(proportion = count / total_pitches)

ggplot(pitch_summary, aes(x = year, y = proportion, fill = pitch_type)) +
  geom_area(position = "stack") +
  labs(title = "Distribution of Pitch Types From 2019-2023",
       x = "Year",
       y = "Proportion of Pitches",
       fill = "Pitch Type") +
  theme_minimal()

burnes_good_pitch_distribution = burnes_good_pitch

# Extract the year from the date column
burnes_good_pitch_distribution$year <- as.numeric(format(as.Date(burnes_good_pitch_distribution$game_date, format = "%Y-%m-%d"), "%Y"))
burnes_good_pitch_distribution = burnes_good_pitch_distribution %>% filter(pitch_type != "")

```

```

# Summarize the data by pitch type and year
good_pitch_sum <- burnes_good_pitch_distribution %>%
  group_by(year, pitch_type) %>%
  summarize(count = n()) %>%
  ungroup()

# Calculate the total pitches per year to get the proportion
good_pitch_sum <- good_pitch_sum %>%
  group_by(year) %>%
  mutate(total_pitches = sum(count)) %>%
  ungroup() %>%
  mutate(proportion = count / total_pitches)

# Plot the distribution of pitches by year
ggplot(good_pitch_sum, aes(x = year, y = proportion, fill = pitch_type)) +
  geom_area(position = "stack") +
  labs(title = "Distribution of Good Pitches From 2019-2023",
       x = "Year",
       y = "Proportion of Pitches",
       fill = "Pitch Type") +
  theme_minimal()

burnes_bad_pitch_distribution = burnes_bad_pitch

# Extract the year from the date column
burnes_bad_pitch_distribution$year <- as.numeric(format(as.Date(burnes_bad_pitch_distribution$game_date, format = "%Y-%m-%d"),
burnes_bad_pitch_distribution = burnes_bad_pitch_distribution %>% filter(pitch_type != "")

# Summarize the data by pitch type and year
bad_pitch_sum <- burnes_bad_pitch_distribution %>%
  group_by(year, pitch_type) %>%
  summarize(count = n()) %>%
  ungroup()

# Calculate the total pitches per year to get the proportion
bad_pitch_sum <- bad_pitch_sum %>%
  group_by(year) %>%
  mutate(total_pitches = sum(count)) %>%
  ungroup() %>%
  mutate(proportion = count / total_pitches)

# Plot the distribution of pitches by year
ggplot(bad_pitch_sum, aes(x = year, y = proportion, fill = pitch_type)) +
  geom_area(position = "stack") +
  labs(title = "Distribution of Bad Pitches from 2019-2023",
       x = "Year",
       y = "Proportion of Pitches",
       fill = "Pitch Type") +
  theme_minimal()

burnes_cutter <- burnes_total %>%
  filter(pitch_type == 'FC')
burnes_cutter_numeric <- sapply(burnes_cutter, is.numeric)
burnes_cutter <- burnes_cutter[, burnes_cutter_numeric]
cor_cutter <- cor(burnes_cutter)

selected_variables <- c("release_speed", "effective_speed", "release_pos_x", "release_pos_y", "release_pos_z", "pfx_x", "pfx_z",
subset_cor_cutter <- cor_cutter[selected_variables,selected_variables]

suppressWarnings(corrplot(
  subset_cor_cutter,
  method = "color",
  tl.col = "black",
  tl.srt = 45,
  addCoef.col = "black",

```

```

col = colorRampPalette(c("#2166ac", "#4393c3", "#92c5de", "#d1e5f0", "#f7f7f7", "#fddbc7", "#f4a582", "#d6604d", "#b2182b"))(10)
number.cex = 0.6))

burnes_changeup <- burnes_total %>%
  filter(pitch_type == 'CH')
burnes_changeup_numeric <- sapply(burnes_changeup, is.numeric)
burnes_changeup <- burnes_changeup[, burnes_changeup_numeric]
cor_changeup <- cor(burnes_changeup)

selected_variables <- c("release_speed", "effective_speed", "release_pos_x", "release_pos_y", "release_pos_z", "pfx_x", "pfx_z", "pfx_y")
subset_cor_changeup <- cor_changeup[selected_variables,selected_variables]

suppressWarnings(corrplot(
  subset_cor_changeup,
  method = "color",
  tl.col = "black",
  tl.srt = 45,
  addCoef.col = "black",
  col = colorRampPalette(c("#2166ac", "#4393c3", "#92c5de", "#d1e5f0", "#f7f7f7", "#fddbc7", "#f4a582", "#d6604d", "#b2182b"))(10)
  number.cex = 0.6))

burnes_curveball <- burnes_total %>%
  filter(pitch_type == 'CU')
burnes_curveball_numeric <- sapply(burnes_curveball, is.numeric)
burnes_curveball <- burnes_curveball[, burnes_curveball_numeric]
cor_curveball <- cor(burnes_curveball)

selected_variables <- c("release_speed", "effective_speed", "release_pos_x", "release_pos_y", "release_pos_z", "pfx_x", "pfx_z", "pfx_y")
subset_cor_curveball <- cor_curveball[selected_variables,selected_variables]

suppressWarnings(corrplot(
  subset_cor_curveball,
  method = "color",
  tl.col = "black",
  tl.srt = 45,
  addCoef.col = "black",
  col = colorRampPalette(c("#2166ac", "#4393c3", "#92c5de", "#d1e5f0", "#f7f7f7", "#fddbc7", "#f4a582", "#d6604d", "#b2182b"))(10)
  number.cex = 0.6))

burnes_sinker <- burnes_total %>%
  filter(pitch_type == 'SI')
burnes_sinker_numeric <- sapply(burnes_sinker, is.numeric)
burnes_sinker <- burnes_sinker[, burnes_sinker_numeric]
cor_sinker <- cor(burnes_sinker)

selected_variables <- c("release_speed", "effective_speed", "release_pos_x", "release_pos_y", "release_pos_z", "pfx_x", "pfx_z", "pfx_y")
subset_cor_sinker <- cor_sinker[selected_variables,selected_variables]

suppressWarnings(corrplot(
  subset_cor_sinker,
  method = "color",
  tl.col = "black",
  tl.srt = 45,
  addCoef.col = "black",
  col = colorRampPalette(c("#2166ac", "#4393c3", "#92c5de", "#d1e5f0", "#f7f7f7", "#fddbc7", "#f4a582", "#d6604d", "#b2182b"))(10)
  number.cex = 0.6))

burnes_slider <- burnes_total %>%
  filter(pitch_type == 'SL')
burnes_slider_numeric <- sapply(burnes_slider, is.numeric)
burnes_slider <- burnes_slider[, burnes_slider_numeric]
cor_slider <- cor(burnes_slider)

selected_variables <- c("release_speed", "effective_speed", "release_pos_x", "release_pos_y", "release_pos_z", "pfx_x", "pfx_z", "pfx_y")
subset_cor_slider <- cor_slider[selected_variables,selected_variables]

```

```
suppressWarnings(corrplot(  
  subset_cor_slider,  
  method = "color",  
  tl.col = "black",  
  tl.srt = 45,  
  addCoef.col = "black",  
  col = colorRampPalette(c("#2166ac", "#4393c3", "#92c5de", "#d1e5f0", "#f7f7f7", "#fddbc7", "#f4a582", "#d6604d", "#b2182b"))(10),  
  number.cex = 0.6))
```