Abstract

The objective of this project is to create an expected run value at the pitch level based on features available in pitch-level Statcast data. This metric will be a context-neutral metric that will attempt to estimate a run value using characteristics available in Statcast data such as spin rate and pitch movement. A model was built for the 2021 season using Random Forest, a supervised Machine Learning Algorithm.

Introduction

Attaching value to a pitch is done through *linear weights*, a class of linear run estimators that are used to determine the relative values of particular events that occur in a baseball game. This was accomplished by first Bill Petti's **baseballr** package and the **scrape_statcast_savant()** function to acquire the pitch-level Statcast data that is publicly available through Baseball Savant for the 2021 season. This gave me an initial dataset of 732,011 observations across approximately 90 different variables.

Under normal circumstances this data would be readily available in a MySQL database where it could be called quickly through a SQL query instead of having to be scraped, but due to limitations in hard drive space, the longer method of scraping the data was chosen.

Data Preparation and Cleaning

The data set contains every pitch for the 2021 Major League Baseball season from Spring Training to the end of the postseason loaded into a data frame. Spring training and postseason data was filtered out so that the dataset only contained pitches from the regular season. Pitcher and batter names, as well as pitcher height were added using Chadwick Bureau's Baseball Databanks and matched using a player's unique **MLBAM ID**.

Vertical Approach Angle, or the angle in which a pitch crosses the plate, was also added using the following code:

```
df <- df %>%
  mutate(vy_f = -sqrt(vy0^2 - (2 * ay *(50 - (17/12)))),
         t = (vy_f-vy0) / ay,
         vz_f = vz0 + (az * t),
         vaa = -atan(vz_f/vy_f) * (180 / pi))

# Where, per Statcast's documentation:

# vy0 = The velocity of the pitch, in feet per second, in y-dimension*, determined at y=50 feet. (*toward home plate)

# ay = The acceleration of the pitch, in feet per second per second, in y-dimension, determined at y=50 feet.# y0 = 50 ("y=50 feet").
# yf = 17/12 (home plate, converted to inches).

# vz0 = The velocity of the pitch, in feet per second, in z-
dimension**, determined at y=50 feet. (**vertically)

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

The next step was to get pitch-level linear weights from pitch-level run expectancy, or the mean number of runs that can be expected to score given specific base, out, and count states. This was an easy task thanks to the **baseballr** package. The run_expectancy_code() and linear_weights_savant() functions where able to do a lot of the heavy lifting to get the run value of every pitch for the season, with a positive value benefitng the hitter, and a negative value benefiting the pitcher.

Event	Linear Weight
Home Run	1.37
Triple	1.06
Double	0.78
Single	0.47
Hit by Pitch	0.39
Balls	0.05
Strikes	-0.05
Outs	-0.24

Choices were made here regarding the linear weights and what to credit pitchers for that can be easily adjusted with future use.

1. Pitches resulting in a walk have the same value as a ball, pitches resulting in a strikeout have the same value as a strike.
2. All strikes are equal, so a swing and miss would be the same value as a called strike.
3. A pitch is credited with the same run value for an out (-0.24) regardless of the out event. For example, if a pitch results in a double play, it will be treated as one out.
4. Outs are credited to a pitch if a fielding error was made.

The values were assigned to every pitch in the data set by event. 2,063 pitch events were dropped due to NAs in the new linear weight colum in order to avoid issues with the Random Forest algorithm. These NAs are for events such as pickoff attempts. After run values were added, additional variables were created for non-fastballs. Four-seam fastballs and sinkers were used as the fastball group, while sliders, curveballs, splitters, knuckle curves, changeups, and cutters were used as the non-fastball group.

Variable	Description
velo_diff	The difference between average fastball velocity and the velocity of non-fastball pitches.
hmov_diff	The difference between average fastball run and the horizontal movement of non-fastball pitches.
vmov_diff	The difference between average fastball ride and the vertical movement of non-fastball pitches.

The final piece of data cleaning involved flipping horizontal release position and horizontal movement for left-handed pitchers so that those two variables were on the same scale as right-handed pitchers.

Feature Selection
For the features selected in the model, fastballs and non-fastballs were grouped again and a Boruta algorithm was applied to tell me the importance of each feature when predicting a pitch's run value. The algorithm was run for six pitch groups (Four-seamers, sinkers, sliders, cutters, curveballs (CB and KC), changeups (CH and FS), with the features and results being stored for later use.

In order to save on processing time these features were

For fastballs (four-seamers, sinkers) the following code was used:

```
Boruta PitchType <- Boruta(lin_weight ~ release_speed +
release_pos_x_adj + release_extension + release_pos_z + pfx_x_adj +
pfx_z + plate_x + plate_z + release_spin_rate + spin_axis + vaa,
data=rr_data sampled)
```

For non-fastballs:

```
Boruta_PitchType <- Boruta(lin_weight ~ release_speed +
release_pos_x_adj + release_extension+ release_pos_z + pfx_x_adj +
pfx_z + plate_x + plate_z + release_spin_rate + spin_axis +
velo_diff + hmov_diff + vmov_diff, data = rr_data_sampled)
```

The features chosen for this model were:

Pitch Type	Features
FF, SI	Release speed, Adjusted vertical release position, horizontal release position, adjusted vertical pitch movement, horizontal pitch movement, release spin rate, release extension, spin axis, vertical approach angle, vertical pitch location, horizontal pitch location
CH, FS, CB, FC	Release speed, Adjusted vertical release position, horizontal release position, adjusted vertical pitch movement, horizontal pitch movement, release spin rate, release extension, spin axis, velocity differential, vertical movement differential, horizontal movement differential, vertical pitch location, horizontal pitch location
SL	Release speed, Adjusted vertical release position, horizontal release position, adjusted vertical pitch movement, horizontal pitch movement, release spin rate, release extension, spin axis, velocity differential, vertical movement differential, horizontal movement differential

Building and Validating the Model
Three functions were built in order to fit, validate and apply the model.

```
rf_model <- function(df, features) {

  features1 <- append(features, c("lin_weight"))
  df1 <- df[,features1]

  randomForest(lin_weight ~ ., data = df1, importance = TRUE,
na.action=na.roughfix)
```

The first function takes in a data frame as well as the features built through the Boruta algorithm and fit to a random forest model, and then returning the model as output. The next function was used to split the data into training (70% of data) and test sets (30% of data) and fit the model along combinations of platoon splits (lefty vs lefty, lefty vs righty, righty vs righty, righty vs lefty).

The model's predictions were then applied to the test set and a Root Mean Squared Error was produced. The validation was performed on the six different pitch type groups (Four-seam fastballs, sinkers, cutters, changeups, sliders, and curveballs). 24 random forest models were created (six different pitch groups times four different platoon combinations), producing a total validation RMSE of **0.10**, or to put in another way, the predicted run value was off from the actual run value by approximately .10 runs.

Pitch Type	RMSE
FF (Four-Seam Fastball)	0.1009
SI (Sinker)	0.0964
FC (Cutter)	0.0967
CH (Changeup and Splitter)	0.0965
SL (Slider)	0.0969
CB (Curveball and Knucklecurve)	0.0857

Results
After validation, the models were applied to the 2021 data set with each pitch-type being passed through a data frame and features list to get final predictions.

Pitch Type	RMSE
FF (Four-Seam Fastball)	0.0988
SI (Sinker)	0.0977
FC (Cutter)	0.0979
CH (Changeup and Splitter)	0.0958
SL (Slider)	0.0965
CB (Curveball and Knucklecurve)	0.0858

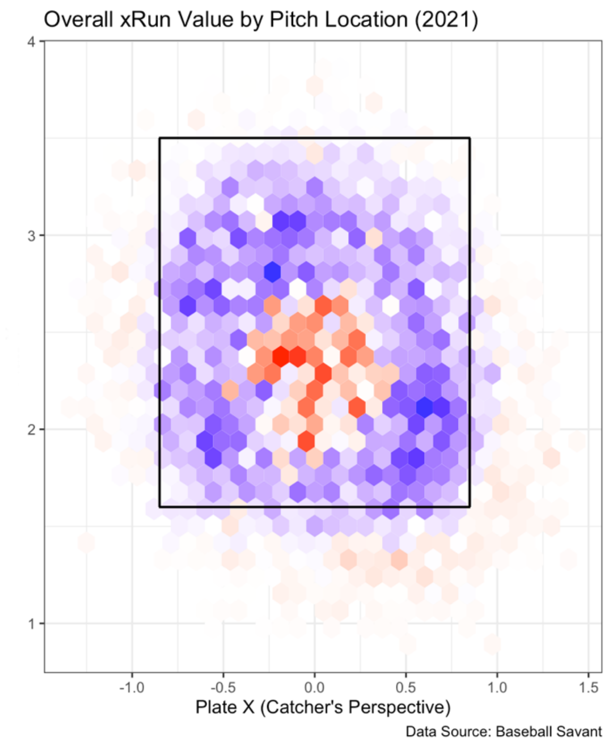
The aggregated RMSE of the predictions came out to **0.10 runs**. These predictions were joined back to the original data frame to be analyzed.

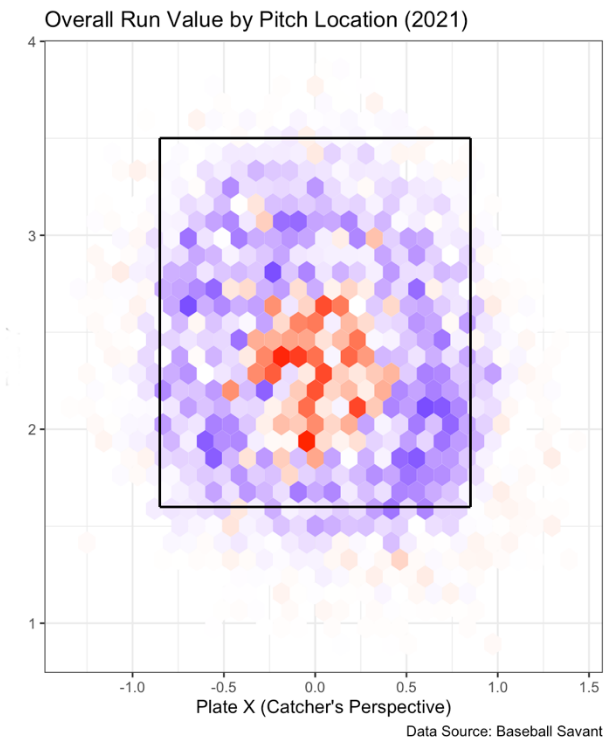
Analysis
The first plot drawn when analyzing how the output of the model was a hex plot. Pitches were grouped by **plate_x** and **plate_y** variables and a run value per 100 pitches was calculated by set of coordinates. The plot was colored by gradient, with blue representing a negative run value (good for the pitcher) and red representing a positive run value (good for the hitter).

The plots themselves worked as a sanity check to some degree. The heart of the strike zone was filled with red which makes sense, this region carried a .396 wOBA (weighted on-base average), which was 130 points higher the outer edges of the strike zone.

What I did like is the model tended to give a little more credit to the pitcher in the upper and lower regions of the zone as evidenced by the deeper blues and lavenders in the top half of the plot. Given what has been discovered regarding Vertical Approach Angle and its relationships to whiffs in the upper parts of the strike zone, it makes sense that the model would have predicted more strikes in these areas.

Factoring this with the incorporation of movement and velocity differentials, it's clear that Expected Run Value model sees these factors as a plus for pitchers. It can be inferred that the model appreciates the notion of high-spin, high-velocity fastballs paired with secondary pitches that differentiate in movement and velocity from the fastball.



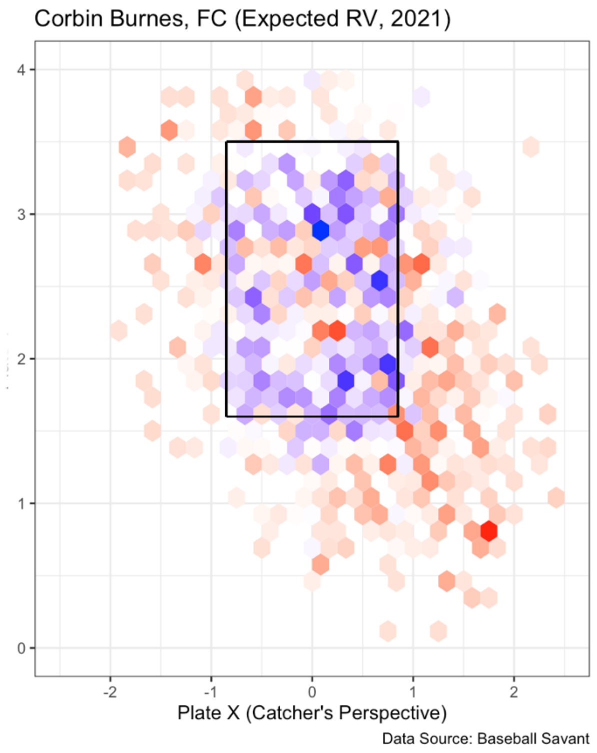


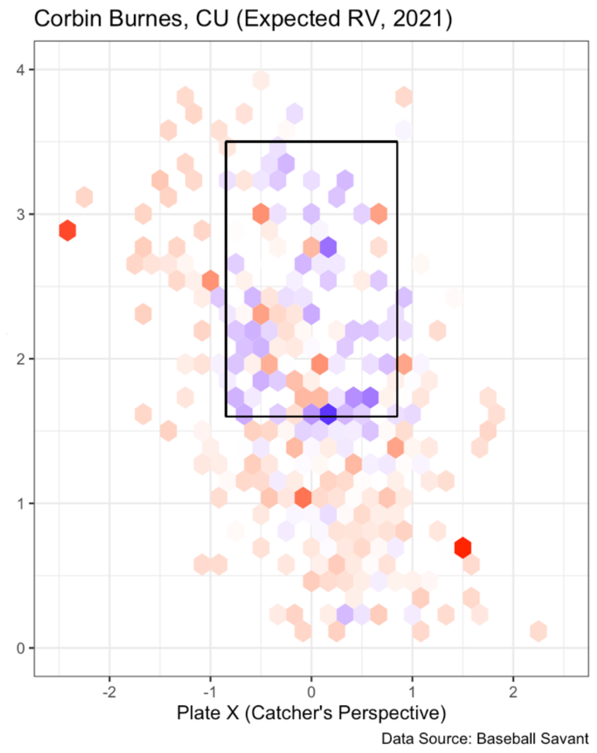
The final plots look at two individual pitchers, Corbin Burnes and Drew Rasmussen. Burnes was chosen due to his impressive 2021 season that saw him finish with 7.5 Wins Above Replacement according to Fangraphs, and a FIP of 1.63. His cutter and curveball according to Baseball Savant's Run Value metric were two of the best pitches in baseball that season. Rasmussen was chosen due to his appearance in the leaderboards below.

The plots were a great tool to visualize a pitcher's arsenal and the expected results pitchers can get in certain areas of the strike-zone. With Corbin Burnes, his cutter reigned supreme throughout most of the zone, showing just how dominate that pitch can be in all areas. The model also predicted negative run values on his curveball in the lower portion of the strike-zone, even predicting some negative values outside of the strike zone, below the knees of the batter.

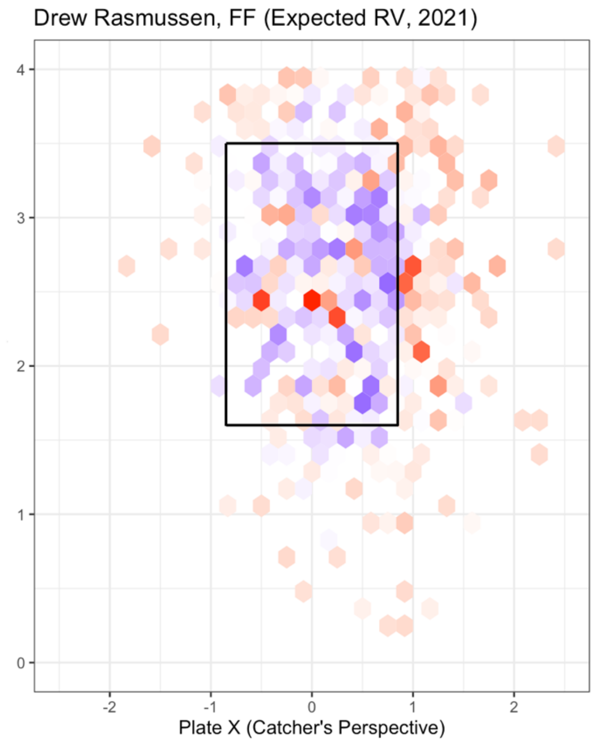
For Rasmussen, the model didn't predict the same amount of domination in the strike-zone as it did for Burnes, but it was still producing interesting nuggets. Again, we can see the model favoring high Vertical Approach Angles and spin when it comes to four-seam fastballs located in the upper parts of the strike zone.

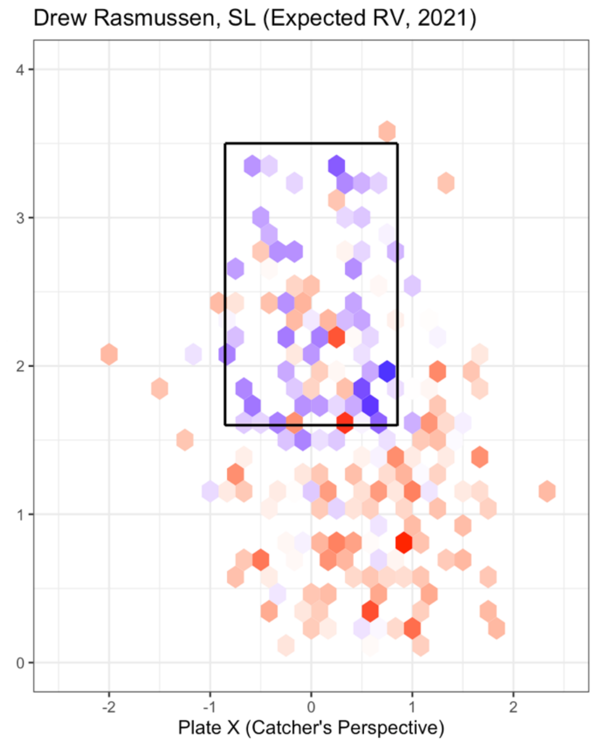
Corbin Burnes, Cutter and Curveball





Drew Rasmussen, Fastball and Slider (2021)





Conclusions

Overall, I found these random forest models to have been a success in what I was trying to accomplish. The goal was to create an expected run value using pitch data available to the public in through Baseball Savant. This pitch level model and it's predictions are a solid context neutral tool that can be used to assess the quality of pitch thrown. There were a few pitfalls and short comings I wanted to address.

1. Due to processing limitations, it was not feasible to evaluate the expected run value on a season-to-season basis. As it stands it is unknown how predictive this metric is. This could be a way to check for an overfitting of the model as well, but random forests tend to not overfit if the data has been pre-processed and cleaned.
2. The model assigned values to each pitch in a vacuum. Things like pitch sequencing, the opposing batter, and game state (score, inning, pitch count) were not accounted for. The model also did not assess contact quality (launch speed, launch angle) for pitches that were put in play.
3. The cost of running these models was hefty, the run time to validate and apply was about 4 hours for a season's worth of data. Running a model using XGBoost feels like a more useful tool, that could even be more accurate at predicting the target variable of run value, and I hope to try it soon.
4. It will be worth re-running the model again with the plate location features being dropped from non-fastballs given the context-neutral nature of the linear weights used. This may give added weight to pitch movement, and movement differentials between fastballs and off-speed pitches. The Boruta algorithm did drop the **plate_x** and **plate_z** variables for sliders on this first go around.
5. There were some factors I wish I could have included, as well as factors that did not work out well. These factors include Alan Nathan's inferred spin calculation, as well as the differential in observed spin to inferred spin in certain pitch pair. I attempted to include estimated arm angle and arm angle differentials but that bore no fruit. The code for both inferred spin and estimated arm angle are still inside the code for this project and is linked below.

Leaderboards

The following leaderboards were put together by pitch group, sorted by Expected Run Value (xRV). xRV is created as a rate stat to be interpreted as runs allowed per 100 pitches. xRV+ is the rescaling of xRV where 100 is average. Pitch level expected wOBA is also included.

Example: *For the 2021 season, Drew Rasmussen has a -0.63 xRV, so we predict that he will prevent 0.63 runs per 100 fastballs thrown on average.*

Leaderboards are min. 500 pitches and sorted by Expected Run Value (xRV).

FOUR SEAM FASTBALL

Pitcher	Count	Run Value	xRV	xRV+	xwOBA
Jose Ruiz	564	-1.46	-0.99	110	.348
Jake McGee	727	-1.56	-0.94	110	.279
Liam Hendricks	653	-0.81	-0.89	109	.282
Drew Steckenrider	573	-1.08	-0.68	109	.296
Drew Rasmussen	651	-0.88	-0.63	108	.317
Chad Green	769	-0.45	-0.61	108	.289
Luis Patiño	758	-0.47	-0.55	108	.338
Jacob deGrom	704	-1.3	-0.54	108	.232
Carlos Rodon	1127	-1.08	-0.5	108	.279
Kevin Gausman	1387	-0.54	-0.5	108	.352

SINKER

Pitcher	Count	Run Value	xRV	xRV+	xwOBA
Josh Hader	553	-1.7	-1.22	109	.228
Brandon Woodruff	688	-1.11	-1.16	108	.342
Sandy Alcantara	832	-1.49	-1.15	108	.308
Zack Wheeler	531	-1.26	-1.11	108	.265
Jonathan Loaisiga	544	-1.47	-1.06	108	.255
Ranger Suárez	622	-1.19	-0.92	108	.243
Tim Mayza	543	-1.65	-0.87	107	.284
Adrian Houser	1131	-1.06	-0.62	107	.303
Adam Wainwright	768	-1.26	-0.56	106	.365
Chris Bassitt	823	-0.74	-0.51	106	.355

CUTTER

Pitcher	Count	Run Value	xRV	xRV+	xwOBA
Kenley Jansen	538	-1.88	-1.24	109	.313
Emmanuel Clase	653	-1.2	-0.71	107	.265
Corbin Burnes	1121	-0.61	-0.51	106	.252
Cal Quantrill	518	-0.63	-0.37	106	.299
Tyler Anderson	640	-0.39	-0.35	106	.281
Walker Buehler	504	-0.44	-0.21	106	.291
Mark Melancon	569	-0.38	-0.12	105	.354
Bryan Shaw	988	-0.27	0.1	105	.337
Adam Wainwright	606	0.18	0.24	104	.342
Tyler Alexander	526	0.58	0.31	104	.291

CHANGEUPS

Pitcher	Count	Run Value	xRV	xRV+	xwOBA
Jordan Montgomery	584	-1.08	-0.68	108	.283
Devin Williams	591	-0.88	-0.49	107	.208
Cole Irvin	528	-0.01	-0.32	107	.308
Trevor Rogers	559	0.9	-0.3	107	.230
Hyun Jin Ryu	659	-0.02	-0.16	106	.318
Sean Manaea	688	-0.75	-0.16	106	.268
Steven Matz	518	-0.5	-0.15	106	.291
Luis Castillo	837	-0.37	-0.09	106	.244
Nabil Crismatt	659	-0.13	-0.07	106	.287
Eduardo Rodriguez	543	0.44	0.02	106	.270

SLIDERS

Pitcher	Count	Run Value	xRV	xRV+	xwOBA
Yu Darvish	565	-0.8	-0.82	106	.269
Lance McCullers Jr.	670	-0.8	-0.66	106	.236
Casey Mize	601	-1.2	-0.63	105	.295
Joe Musgrove	721	-0.74	-0.62	105	.261
Sandy Alcantara	692	-1.08	-0.62	105	.244
Ervin Santana	520	-1.08	-0.61	105	.281
Diego Castillo	564	-1.01	-0.56	105	.206
Lucas Giolito	580	-0.65	-0.38	105	.235
Carlos Rodón	548	-0.54	-0.37	105	.160
Jon Gray	865	-0.29	-0.3	105	.234

CURVEBALLS

Pitcher	Count	Run Value	xRV	xRV+	xwOBA
Julio Urías	828	-1.67	-1.45	110	.206
Adam Wainwright	957	-1.22	-1.05	109	.279
Charlie Morton	990	-1.19	-0.97	109	.227
Joe Musgrove	607	-1.25	-0.88	108	.225
Lance McCullers Jr.	624	-0.92	-0.65	108	.269
Nathan Eovaldi	518	-1.02	-0.58	107	.223
Rich Hill	836	-0.59	-0.34	107	.322
Jordan Montgomery	569	-0.33	-0.13	106	.216
Max Fried	612	-0.4	-0.05	106	.203
Framber Valdez	591	-0.13	-0.02	106	.229

References

The code for this project can be found on my GitHub here: https://github.com/christopherjluke/baseball_projects/blob/main/pitch_model_v01.R

Slowinski, P. (2010, February 25). Linear Weights. Retrieved September 16, 2022, from Fangraphs website: https://library.fangraphs.com/principles/linear-weights/

Chamberlain, A. (2022, February 1). A Visualized Primer on Vertical Approach Angle (VAA). Retrieved September 16, 2022, from Fangraphs website: https://blogs.fangraphs.com/a-visualized-primer-on-vertical-approach-angle-vaa/

Gilani, S. and Petti, R. (2021). baseballr: The SportsDataverse's R Package for Baseball Data. Retrieved September 16, 2022, website: https://billpetti.github.io/baseballr/index.html

Slowinski, P. (2010, February 15). wOBA I Sabermetrics Library. Retrieved September 16. 2022, from Fangraphs website: https://library.fangraphs.com/offense/woba/

Bentéjac, C. & Csörgő, A. & Martínez-Muñoz, G., (2019). A Comparative Analysis of XGBoost, from website: https://www.researchgate.net/publication/337048557_A_Comparative_Analysis_of_XGBoost