Atlanta Braves R&D Question Responses

Jackson Balch

Sep 27, 2023

1. On 8/24/2021, the Cardinals trailed the Tigers 4-3 going into the top of the 9th. To begin this inning, Daz Cameron doubled, Akil Baddoo struck out, and Jonathan Schoop grounded out, moving Cameron to 3rd. The batter is now Robbie Grossman. Assume Luis Garcia will pitch through the 5th spot in the batting order, Jeimer Candelario. Should the Cardinals intentionally walk Grossman? Describe what your process would be to determine whether to pitch to him. The following link contains the box score information for this game: https://www.mlb.com/gameday/tigers-vs-cardinals/2021/08/24/632781#game_state=final,lock_state=final,game_tab=box,game=63278

At this point, there is now two outs with a runner on third. Using FanGraphs RE24 Matrix from 2014, the run expectancy of this situation is 0.413. If Grossman is walked, the run expectancy shifts to 0.471, meaning the Cardinals give up 0.058 expected runs by walking Grossman. This is not a high enough

In this at bat, there are only a few possible outcomes that actively harm us. Any hit by Grossman likely scores Cameron. Because there are two outs, a pop-up does not allow for a tag, a strikeout ends the inning regardless, and a walk just advances us to the next scenario. Really, the only thing we are worried about is Grossman reaching base safely or homering. This probability can be somewhat estimated by Grossman's OBP or WOBA. Grossman's WOBA was 0.337 in 2021 according to statcast, which ranked him at 86th in the majors out of 132 qualified hitters according to FanGraphs. This is a below average hitter, which does not make sense to walk ever unless you are absolutely certain that you have a better chance to get the next batter out. This would set the situation the same, with any hit by Jeimer Candelario being a repeat of the scenario previously outlined with a hopefully worse probability. So, is Jeimer Candelario worse than Robbie Grossman? No. Candelario had a 0.344 WOBA and ranked 44th, which is better than Grossman. It makes almost no sense to me to walk Grossman, like the Cardinals did in actuality. I would have chosen to pitch to Grossman.

2. You are running a generic mid-market team and are exploring the idea of signing Aaron Judge this offseason. What contract would you be willing to offer him? Please explain your thought process and discuss any important considerations.

I'm going to start under the assumption that my team wants to sign Judge, and you're asking what a reasonable competitive offer would be to get him to sign. Assuming Judge would not take less than the Yankees offer, any competitive offer starts at over $40,000,000 annually. That being said, to use the average budget as my own, that gives me a total of about $160,000,000, or $120,000,000 after Judge signs at at least $40 Million annually. In order to entice Judge, I

probably need to add an additional bump. Using Bryce Harper as a comparable for a top free-agent Outfielder, he went from about $21 Million/year in his final deal with the Nationals to $25 Million/year in his Phillies deal. That's about a 19% bump, which in Judge's case would bring his valuation to $47 Million annually. Judge is older than Harper was, but I would still reason that even at age 31, it would take at least a 9 year, $423 Million deal to pry Judge away from the Yankees, leaving me at about $115 Million annually.

That's an unreasonable amount to run off and afford a competitive roster on other parts of the roster, unless I have a great farm system and believe in my prospects. In that case, I now have the remaining budget of a team like the Diamondbacks, Reds, Guardians, and about $30 Million more than the Rays. If I'm a smart team and great at hitting on prospects, I can afford that to help maximize a championship window.

In this scenario, I probably would not be willing to offer anything over $45 Million at a low term, especially if I'm within a championship window. While that may feel like still a massive deal, especially for a mid-market team, that deal allows me an added business opportunity to offset that increase in budget by adding a star player like Judge. Even removing any talent or on-field play, adding a superstar like Judge gives me extra national TV games, boosted jersey and ticket sales, and gets my team in the national spotlight. The team becomes a free agent destination to be able to play with Judge, they get more media time, they get a larger share of conversation and attention, and overall, my team becomes worth more by paying more. From a pure business standpoint, I would feel comfortable offering Judge a $135 Million deal over three years. That's $15 Million more than the Yankees offer over that term with a chance to chase a ring, and with an opportunity for him to also still cash out on another megadeal at age 34. This also helps to make up for Judge with the personal marketing value lost by moving to a mid-market team, while also giving him flexibility as to where he wants to finish his career. Judge is also now able to help boost my championship odds while not hamstringing long-term cap.

3. Pitcher A walks half the batters he faces and strikes out the other half. Pitcher B doesn't walk or strike out any of the batters he faces. Which pitcher would you prefer? What ratio of strikeouts to walks would make you indifferent between the two pitchers?

I would prefer pitcher A for two reasons: their durability and effectiveness.

To start, pitcher A would be a decently effective pitcher. This situation can be modeled by the negative binomial distribution nb(x: 3, 0.5). This is equivalent to $\binom{x+3-1}{3-1}0.5^3(1-0.5)^x$ where x is the number of batters faced, 0.5 is the probability of a strikeout, and we need three strikeouts to end an inning.
The probability of giving up zero runs in inning, A.K.A the pitcher striking out three batters out of six, could be modeled by $\sum_{i=0}^{3}\binom{6-i-1}{2}0.5^3 0.5^{3-i}$, or

$\binom{6-1}{3-1}0.5^3 0.5^3 + \binom{5-1}{3-1}0.5^3 0.5^2 + \binom{4-1}{3-1}0.5^3 0.5^1 + \binom{3-1}{3-1}0.5^3 = 0.15625 + 0.1875 + 0.1875 + 0.125 = 0.65625$. This means that there is a 65.625% chance of a scoreless inning.

To model only 1 run scored, we would add on the probability of one run scoring, getting up to 0.7734375, or 77.34% chance of a one run or less inning. In the average inning, they get by without giving up a run.

I would feel confident with having a pitcher I believed could escape with a run or less in most innings. This is an effective closer in most games, and I feel comfortable with this pitcher to throw most of most games. That's a great pitcher that I feel good about in my rotation, as compared to someone who is unknown.

While maybe outside the bounds of realism, Pitcher A also has essentially unlimited durability and usage in this scenario. In this hypothetical, I can pitch pitcher A every day for a full nine innings if I want to, without him losing any of his efficiency, knowing that he will still either walk or strike out a runner at p = 0.5. In a blowout, I don't have to waste the bullpen, or in the case of injuries, I know this pitcher will never lose efficiency even if hurt. This is extremely valuable to have essentially an infinite pitcher, especially if good.

Without knowing more about Pitcher B, it's hard to say that they're a bad pitcher. I don't know how effective they are at creating ground balls or flyouts, so it's hard to judge what the probability that a runner gets out at any given moment is. While there is the novel idea of just never swinging at a pitcher that can neither walk me nor strike me out and breaking the game, Pitcher A is just such a safer, known pitcher that I would feel incredibly comfortable with having on my team for their effectiveness and their durability.

I would probably feel comfortable with Pitcher A at a ratio anywhere above a 2:3 ratio of strikeouts to walks. P = 0.4 gets to the probability of a scoreless inning now being only 45.56%, with a one-run or less being 58% chance.

4. Briefly explain how you would go about estimating the effect of catcher framing at the major league level? Assume you only have access to the identities of the people involved, information about the pitch (location, characteristics, etc.), and information about the game (count, inning, score, etc.).

In order to estimate the effect of catcher framing at the major league level, I would likely start by developing metrics surrounding the expected value of a pitch in order to attribute the sum of the difference between actual and expected values to the catcher.

I start by determining the expected value of a pitch that wasn't swung at. For each pitch type and pitcher and batter handedness, I find and model the expected value of a pitch. For the model's simplicity sake, I assume that all umpires, catchers, and pitchers are equivalent at this point, and that no pitcher or catcher implicitly has a bias towards getting a ball or strike call. This

value would be between 0 and 1, where 1 is always called a strike and 0 is always a ball. I would either develop a simple expected value of each pitch by taking the ratio of called strikes and the number of all pitches for each type and hand in a binned area or develop generalized linear models for each quadrant of the strike zone. Developing a model for each quadrant would allow for a general umpire's tendency to call strikes in a particular location to be taken into effect. The simple model would obviously be much less computationally expensive, and I would look for efficiency in that and compare it to the efficiency of a more robust model. For example with the simple model, if in one area of one quadrant there are 60/80 fastballs from right-handed pitchers against lefties that are called strikes, the expected value would be 0.75 for that particular pitch.

I then attribute to the catcher the difference between the actual value (0 if called a ball, 1 if called a strike) and the expected value. For the previous example, if that pitch was actually called as a strike, then the catcher would receive 0.25 strikes of credit. I would likely start by totalling the "strikes stolen" by a catcher.

This would likely need to be further transformed to be a fair metric by only using pitches where the expected value is not the same as the actual value in order to help allow for catchers to not be penalized for catching for pitchers who generally throw high numbers of true strikes. It would be unfair to penalize a catcher's average expected strikes stolen if they generally catch a high number of strikes that are unquestionably strikes. This would also make most sense to compare catchers on a pitch-by-pitch basis, so this further normalized "strikes stolen per questionable pitch", or SSQP, would be the metric I would use to compare catchers. This would allow me to be able to isolate the most effective catchers at framing, and I could also find group statistics like mean and median for this statistic. This statistic could allow me to figure out what the general effect of framing is as a whole to the league.

Braves Modeling

Jackson Balch

2023-09-27

Modeling Questions

Question 1

1. Create TWO models to predict the likelihood of a swing and miss based on the characteristics of a curveball. Evaluate and compare the performances of your models using any method(s) you'd prefer. Explain your results in 500 words or less.

Whenever I get any data, my first goal is to examine it and start to try to draw some basic conclusions from an overview. This involves some quick cleaning, and for a dataset of this size, that means just removing data that doesn't make sense. In this case, I'm removing any rows with NA values, or that are negative feet above the ground when crossing the plate since it doesn't make sense to have data involving balls underground. I assume these to be mistakes in the tracking software, and by looking at the rest of the data, I feel it all tends to be within bounds of realism. In a more complex model or one that I dedicated more time to, I would do my best to find a better way to check these by doing some of my own calculations, but for a short project, I'll continue under the assumptions everything else is valid.

```
data <- read.csv("PitchData.csv")
summary(data)
```

```
##    Pitcher_ID       Pitcher          Pitcher_Throws       Batter_ID
##  Min.   :434671   Length:24128       Length:24128       Min.   :400284
##  1st Qu.:527054   Class :character   Class :character   1st Qu.:502110
##  Median :592314   Mode  :character   Mode  :character   Median :571448
##  Mean   :569342                                         Mean   :552791
##  3rd Qu.:621345                                         3rd Qu.:605141
##  Max.   :669060                                         Max.   :670950
##
##     Batter          Batter_Hits         Game_Date           Top_Bot
##  Length:24128       Length:24128       Length:24128       Min.   :1.000
##  Class :character   Class :character   Class :character   1st Qu.:1.000
##  Mode  :character   Mode  :character   Mode  :character   Median :1.000
##                                                           Mean   :1.492
##                                                           3rd Qu.:2.000
##                                                           Max.   :2.000
##
##      Inning           Balls           Strikes            Outs
##  Min.   : 1.000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.: 3.000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median : 5.000   Median :1.0000   Median :1.0000   Median :1.0000
```

```
##  Mean   : 5.055   Mean   :0.9089   Mean   :0.8834   Mean   :0.9853
##  3rd Qu.: 7.000   3rd Qu.:2.0000   3rd Qu.:2.0000   3rd Qu.:2.0000
##  Max.   :15.000   Max.   :3.0000   Max.   :2.0000   Max.   :2.0000
##                   NA's   :21       NA's   :21       NA's   :21
##  Pitch_Outcome       Pitch_Type        release_speed      x_movement
##  Length:24128       Length:24128       Min.   :  0.00   Min.   :-14.466
##  Class :character   Class :character   1st Qu.: 85.08   1st Qu.: -6.317
##  Mode  :character   Mode  :character   Median : 90.38   Median : -1.499
##                                        Mean   : 88.85   Mean   : -1.458
##                                        3rd Qu.: 93.80   3rd Qu.:  3.229
##                                        Max.   :146.93   Max.   : 11.857
##
##    z_movement      release_spin_rate    spin_dir     release_pos_x
##  Min.   :-13.558   Min.   :   0      Min.   :  0.0   Min.   :-3.0898
##  1st Qu.:  1.488   1st Qu.:2103      1st Qu.:146.5   1st Qu.:-1.6165
##  Median :  6.042   Median :2270      Median :193.4   Median : 1.6791
##  Mean   :  4.593   Mean   :2191      Mean   :183.5   Mean   : 0.6939
##  3rd Qu.:  8.562   3rd Qu.:2454      3rd Qu.:226.2   3rd Qu.: 2.1688
##  Max.   : 13.358   Max.   :3496      Max.   :359.8   Max.   : 4.6258
##                    NA's   :18
##  release_pos_z   release_extension    plate_x            plate_z
##  Min.   :0.000   Min.   :0.000     Min.   :-3.947303   Min.   :-2.267
##  1st Qu.:5.467   1st Qu.:5.544     1st Qu.:-0.619640   1st Qu.: 1.631
##  Median :5.927   Median :5.887     Median :-0.008070   Median : 2.246
##  Mean   :5.837   Mean   :5.830     Mean   :-0.008552   Mean   : 2.245
##  3rd Qu.:6.274   3rd Qu.:6.146     3rd Qu.: 0.590496   3rd Qu.: 2.865
##  Max.   :7.106   Max.   :7.623     Max.   : 9.670856   Max.   : 7.496
##
```

```
data <- data %>%
  filter(!is.na(Balls)) %>%
  filter(!is.na(release_spin_rate)) %>%
  filter(plate_z >= 0) %>%
  filter(Pitch_Outcome %in% c("StrikeSwinging", "InPlay", "FoulBall"))
data$Pitch_Outcome <- ifelse(data$Pitch_Outcome == "StrikeSwinging", "Miss",
"Contact")
summary(data)
```

```
##    Pitcher_ID       Pitcher          Pitcher_Throws       Batter_ID
##  Min.   :434671   Length:10955       Length:10955       Min.   :400284
##  1st Qu.:527054   Class :character   Class :character   1st Qu.:502110
##  Median :592314   Mode  :character   Mode  :character   Median :571448
##  Mean   :568861                                         Mean   :552632
##  3rd Qu.:621345                                         3rd Qu.:605141
##  Max.   :669060                                         Max.   :670950
##     Batter          Batter_Hits         Game_Date            Top_Bot
##  Length:10955       Length:10955       Length:10955       Min.   :1.000
##  Class :character   Class :character   Class :character   1st Qu.:1.000
##  Mode  :character   Mode  :character   Mode  :character   Median :1.000
##                                                           Mean   :1.489
```

```
##                                                          3rd Qu.:2.000
##                                                          Max.   :2.000
##      Inning           Balls          Strikes           Outs
##  Min.   : 1.000   Min.   :0.000   Min.   :0.000   Min.   :0.0000
##  1st Qu.: 3.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.0000
##  Median : 5.000   Median :1.000   Median :1.000   Median :1.0000
##  Mean   : 5.148   Mean   :1.079   Mean   :1.097   Mean   :0.9796
##  3rd Qu.: 7.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.0000
##  Max.   :15.000   Max.   :3.000   Max.   :2.000   Max.   :2.0000
##  Pitch_Outcome       Pitch_Type        release_speed      x_movement
##  Length:10955       Length:10955       Min.   :  0.00   Min.   :-14.173
##  Class :character   Class :character   1st Qu.: 85.52   1st Qu.: -6.390
##  Mode  :character   Mode  :character   Median : 90.57   Median : -1.406
##                                        Mean   : 89.15   Mean   : -1.470
##                                        3rd Qu.: 93.93   3rd Qu.:  3.198
##                                        Max.   :100.08   Max.   : 11.644
##    z_movement      release_spin_rate    spin_dir     release_pos_x
##  Min.   :-13.558   Min.   :   0      Min.   :  0.0   Min.   :-2.9200
##  1st Qu.:  1.841   1st Qu.:2093      1st Qu.:148.4   1st Qu.:-1.6150
##  Median :  5.984   Median :2266      Median :193.1   Median : 1.6699
##  Mean   :  4.755   Mean   :2182      Mean   :184.0   Mean   : 0.6963
##  3rd Qu.:  8.514   3rd Qu.:2451      3rd Qu.:226.1   3rd Qu.: 2.1659
##  Max.   : 13.213   Max.   :3496      Max.   :359.8   Max.   : 4.5327
##  release_pos_z   release_extension    plate_x             plate_z
##  Min.   :0.000   Min.   :0.000     Min.   :-2.566533   Min.   :0.000
##  1st Qu.:5.464   1st Qu.:5.562     1st Qu.:-0.449811   1st Qu.:1.811
##  Median :5.921   Median :5.895     Median :-0.004943   Median :2.310
##  Mean   :5.832   Mean   :5.837     Mean   :-0.012978   Mean   :2.302
##  3rd Qu.:6.269   3rd Qu.:6.147     3rd Qu.: 0.430871   3rd Qu.:2.804
##  Max.   :7.049   Max.   :7.435     Max.   : 2.295108   Max.   :4.760
```

My first model I'll utilize is a LASSO regression for a generalized linear model. I'll filter down to only use pitches that were recorded as curveballs, and then select only numerical data.

```
set.seed(1)
use_data <- data %>%
  filter(Pitch_Type == "Curveball") %>%
  select(-c(Pitcher, Batter, Pitcher_ID, Batter_ID, Pitch_Type, Game_Date))
cov_mat <- cov(use_data %>% select(-c(Pitch_Outcome)) %>%
select_if(is.numeric))
cov_mat
```

```
##                      Top_Bot       Inning        Balls       Strikes
## Top_Bot           0.247644638  -0.03830886 -0.002119887   0.008217908
## Inning           -0.038308861   6.32977747  0.085697214   0.120236035
## Balls            -0.002119887   0.08569721  0.753443295   0.312029810
## Strikes           0.008217908   0.12023604  0.312029810   0.667929837
## Outs             -0.011588796   0.04769624  0.011609483  -0.012120593
## release_speed    -0.133812289   2.11655541  0.133743947   0.242067000
```

```
## x_movement        -0.415306386  -1.83818796 -0.039747387  -0.476138118
## z_movement        -0.143581512   0.81652785 -0.267328050  -0.667796303
## release_spin_rate -2.856367314 -13.45242434 48.024878202 115.294970142
## spin_dir           7.747343219  18.70388033  3.256822460  19.634288517
## release_pos_x     -0.102860677  -0.02085850  0.005828625  -0.237051714
## release_pos_z      0.030912696  -0.40505727 -0.011869381   0.031938956
## release_extension  0.032478554   0.24416022  0.024731512   0.015366435
## plate_x            0.007880612  -0.06147730 -0.029316286  -0.029662455
## plate_z           -0.001656234  -0.09581624  0.023941834  -0.041811962
##                          Outs release_speed    x_movement    z_movement
## Top_Bot           -0.011588796   -0.13381229   -0.41530639   -0.14358151
## Inning             0.047696236    2.11655541   -1.83818796    0.81652785
## Balls              0.011609483    0.13374395   -0.03974739   -0.26732805
## Strikes           -0.012120593    0.24206700   -0.47613812   -0.66779630
## Outs               0.681372012    0.24855405   -0.16061223   -0.04104340
## release_speed      0.248554048   11.19230286    3.16009484    5.69500614
## x_movement        -0.160612226    3.16009484   25.78184054    8.63383704
## z_movement        -0.041043404    5.69500614    8.63383704   12.79683079
## release_spin_rate  0.579453200 -309.67288387 -317.00345972 -872.44042545
## spin_dir           0.753321301 -146.01906784 -591.59248938 -270.61104264
## release_pos_x      0.010048790    1.76835196    8.07964441    3.46721657
## release_pos_z     -0.009775112    0.19343200   -0.85280056   -0.78106079
## release_extension  0.015526336   -0.03957672    0.19647772   -0.10195282
## plate_x            0.004098345   -0.23030616   -0.53384472   -0.09174495
## plate_z           -0.006572506   -0.03696583   -0.10720855    0.09376255
##                   release_spin_rate      spin_dir release_pos_x
release_pos_z
## Top_Bot               -2.856367e+00     7.7473432 -1.028607e-01
0.030912696
## Inning                -1.345242e+01    18.7038803 -2.085850e-02  -
0.405057274
## Balls                  4.802488e+01     3.2568225  5.828625e-03  -
0.011869381
## Strikes                1.152950e+02    19.6342885 -2.370517e-01
0.031938956
## Outs                   5.794532e-01     0.7533213  1.004879e-02  -
0.009775112
## release_speed         -3.096729e+02  -146.0190678  1.768352e+00
0.193432002
## x_movement            -3.170035e+02  -591.5924894  8.079644e+00  -
0.852800555
## z_movement            -8.724404e+02  -270.6110426  3.467217e+00  -
0.781060790
## release_spin_rate      3.702904e+05 18260.6893936 -1.680158e+02
41.206414444
## spin_dir               1.826069e+04 17256.5146377 -2.237873e+02
20.433624364
## release_pos_x         -1.680158e+02  -223.7872814  3.321233e+00  -
0.382009925
## release_pos_z          4.120641e+01    20.4336244 -3.820099e-01
```

```
0.256876963
## release_extension     -3.110764e+01   -11.7535807  1.887462e-01  -
0.078675760
## plate_x               -1.623773e+01     8.2510082 -1.377873e-01
0.026008827
## plate_z               -2.437691e+01     1.1104754 -1.566958e-02
0.019396689
##                   release_extension       plate_x       plate_z
## Top_Bot                  0.03247855   0.007880612  -0.001656234
## Inning                   0.24416022  -0.061477304  -0.095816240
## Balls                    0.02473151  -0.029316286   0.023941834
## Strikes                  0.01536643  -0.029662455  -0.041811962
## Outs                     0.01552634   0.004098345  -0.006572506
## release_speed           -0.03957672  -0.230306157  -0.036965831
## x_movement               0.19647772  -0.533844724  -0.107208549
## z_movement              -0.10195282  -0.091744955   0.093762552
## release_spin_rate      -31.10763592 -16.237734342 -24.376910418
## spin_dir               -11.75358072   8.251008188   1.110475367
## release_pos_x            0.18874622  -0.137787305  -0.015669585
## release_pos_z           -0.07867576   0.026008827   0.019396689
## release_extension        0.22464965  -0.009054410  -0.015046710
## plate_x                 -0.00905441   0.366633906   0.025437452
## plate_z                 -0.01504671   0.025437452   0.464004213
```

I examined the covariance of my data and decided to remove movement as a variable. While movement may seem like an important piece to determining the swing/miss likelihood of a pitch, that should be captured by release speed, spin angle, and spin rate. Those three variates combine to determine movement along with gravity, so movement's prescence should still be felt within the model and remove some of the covariance.

I then will use a cross-validation training/testing method while finding the optimal penalty value from the LASSO procedure.

```
use_data <- data %>%
  filter(Pitch_Type == "Curveball") %>%
  select(-c(Pitcher, Batter, Pitcher_ID, Batter_ID, Pitch_Type, Game_Date,
x_movement, z_movement))

partition <- createDataPartition(use_data$Pitch_Outcome, p=0.7, list=F)

train <- use_data[partition,]
test <- use_data[-partition,]

myControl <- trainControl(
                          method = "cv", number = 10,
                          summaryFunction = twoClassSummary,
                          classProbs = TRUE
                         )

lasso_tune_grid= expand.grid(alpha = 1, lambda = seq(0.0001, 10, length =
```

```
100))


lasso <- train(Pitch_Outcome ~ ., data = train, method='glmnet', tune_grid =
lasso_tune_grid, trControl=myControl, preProcess = "scale")
```

This was my optimal penalty value.

```
lasso$bestTune$lambda

## [1] 0.04079235

confus <- confusionMatrix(as.factor(predict(lasso, train)),
as.factor(train$Pitch_Outcome), mode="everything")
confus$table

##           Reference
## Prediction Contact Miss
##    Contact     408  129
##    Miss         26   73

confus$byClass['F1']

##        F1
## 0.8403708
```

On the training set, the model had an F1 score of 0.84, which is a great score. This is not predicting either outcome with an overwhelming bias, and seems to fit an appropriate level of specificity and sensitivity.

```
confus_test <- confusionMatrix(as.factor(predict(lasso, test)),
as.factor(test$Pitch_Outcome), mode="everything")
confus_test$table

##           Reference
## Prediction Contact Miss
##    Contact     172   50
##    Miss         13   36

confus_test$byClass['F1']

##        F1
## 0.8452088
```

On the test set, the model appeared to have about an equivalent F1 score, which means it well fit to the training set without overfitting. This is a solid model.

For the second model, I want to utilzie a non-parametric approach. This is an unbalanced dataset, with over three times as many contacts as misses. Thus, a non-parametric model that does not assume normality could apply well here.

```r
use_data %>%
  group_by(Pitch_Outcome) %>%
  summarise(n_rows = length(Pitch_Outcome))
```

```
## # A tibble: 2 × 2
##   Pitch_Outcome n_rows
##   <chr>          <int>
## 1 Contact          619
## 2 Miss             288
```

This is again a generalized linear model predicting the probability of contact or a whiff using a binomial family distribution and generalized-cross-validation.

```r
np_train <- train
np_train$Pitcher_Throws <- ifelse(np_train$Pitcher_Throws == "R", 1, 0)
np_train$Batter_Hits <- ifelse(np_train$Batter_Hits == "R", 1, 0)
np_train$Pitch_Outcome <- ifelse(np_train$Pitch_Outcome == "Miss", 0, 1)

np_model <- gsm(Pitch_Outcome ~ ., family = "binomial", data = np_train,
method="GCV")
pred_values_train <- ifelse(predict(np_model, np_train, type='response') >=
0.5, "Hit", "Miss")
real_vals_train <- ifelse(np_train$Pitch_Outcome > 0.5, "Hit", "Miss")



np_test <- test
np_test$Pitcher_Throws <- ifelse(np_test$Pitcher_Throws == "R", 1, 0)
np_test$Batter_Hits <- ifelse(np_test$Batter_Hits == "R", 1, 0)
np_test$Pitch_Outcome <- ifelse(np_test$Pitch_Outcome == "Miss", 0, 1)

pred_values_test <- ifelse(predict(np_model, np_test, type='response') > 0.5,
"Hit", "Miss")
real_vals_test <- ifelse(np_test$Pitch_Outcome > 0.5, "Hit", "Miss")
```

On the training set:

```r
confus <- confusionMatrix(as.factor(pred_values_train),
as.factor(real_vals_train), mode="everything")
confus$table
```

```
##            Reference
## Prediction Hit Miss
##       Hit  394  101
##       Miss  40  101
```

```r
confus$byClass['F1']
```

```
##        F1
## 0.8482239
```

On the test set:

```
confus_test <- confusionMatrix(as.factor(pred_values_test),
as.factor(real_vals_test), mode="everything")
confus_test$table
```

```
##           Reference
## Prediction Hit Miss
##       Hit  165   41
##       Miss  20   45
```

```
confus_test$byClass['F1']
```

```
##        F1
## 0.8439898
```

This model, similar to above, again appears to be properly fit and not overfitting.

```
dumb_ll <- -log(1/2)
lasso_ll <- LogLoss(predict(lasso, test, type='prob')$Contact,
ifelse(test$Pitch_Outcome == "Miss", 0, 1))
np_ll <- LogLoss(predict(np_model, np_test, type='response'),
np_test$Pitch_Outcome)
```

```
dumb_ll
```

```
## [1] 0.6931472
```

```
lasso_ll
```

```
## [1] 0.5415164
```

```
np_ll
```

```
## [1] 0.5104768
```

To also compare logloss of both models, they both are similar and better than a baseline guess should predict. The second has very similar performance metrics, with me being able to say a lot of the same things from above. Because the second model is a bit more complicated to explain to those unfamiliar with statistical theory, I will choose to use the first model going forward.

Question 2

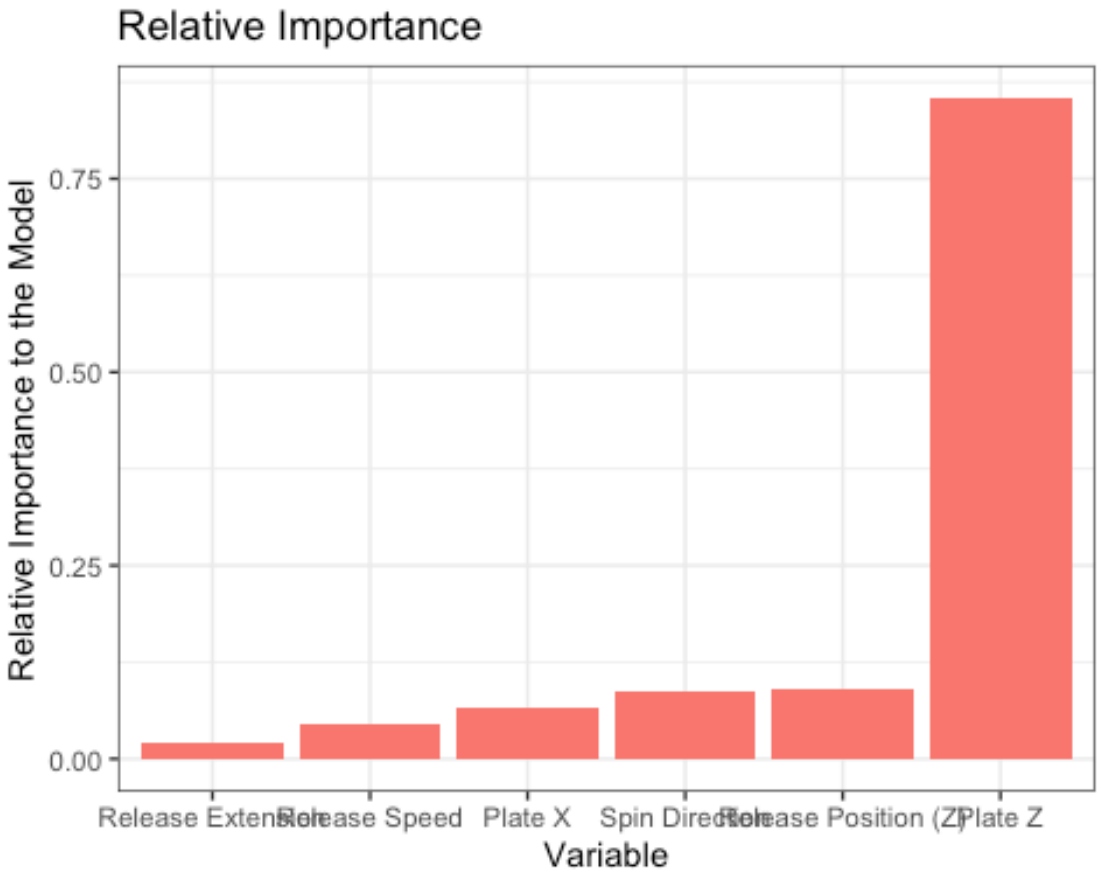
2. Using your preferred model from Question #3, create a visualization to display the most important characteristics of a curveball in recording a swing-and-miss. Explain your visualization in 500 words or less.

```
coefs <- predict(lasso$finalModel, type="coef", s=lasso$bestTune$lambda)
coefs_vals <- coefs[coefs!=0]
coefs_vals <- coefs_vals[2:length(coefs_vals)]
names(coefs_vals) <- c("Release Speed", "Spin Direction", "Release Position
(Z)", "Release Extension", "Plate X", "Plate Z")
coef_df <- data_frame("Variable"=names(coefs_vals), "Value"=coefs_vals)
```

```r
##plot1 <- ggplot(coef_df, aes(x=Variable, y=Value, fill =
as.factor(Variable))) + geom_bar(stat='identity') + theme_bw() +
theme(legend.position = "none") + ggtitle("Coefficient Value")

var_impo <- varImp(lasso$finalModel) %>%
  filter(Overall > 0)

coef_df$importance <- var_impo$Overall
plot2 <- coef_df %>%
  arrange(importance) %>%
  mutate(Variable = factor(Variable, levels = Variable)) %>%
  ggplot(aes(x=Variable, y=importance, fill = "black")) +
  geom_bar(stat='identity') +
  theme_bw() +
  theme(legend.position = "none") +
  ggtitle("Relative Importance") + ylab("Relative Importance to the Model")
grid.arrange(plot2)
```



This visualization is designed to show the relative importance to the model previously built to predict contact vs whiffs on curveball pitches. It's straightforward as easy to read, showing that the Z-location of a curveball is the most important piece of determining whether or not a batter whiffs on a curveball they swing at. The Z-location of a pitch is the height of the ball above the ground when the ball goes over the plate. The coefficient for

this was negative, meaning that hitters tended to swing at high curveballs and whiff as compared to lower curves that they were able to connect with easier. This is followed by release position on the y-axis, which is the height at which the pitcher releases the ball. This coefficient was also negative, meaning higher releases tended to result in higher whiff rates. This then followed by spin direction or spin angle. A positive coefficient here meant that lower angles resulted in more difficult to hit curveballs. In accordance to this model and graph, the hardest curveballs with the highest likelihood to whiff on would be high curveballs with a high release and low spin angle.