M158 Project Part 3 (Elliott, Choi)

Riley and Anna

2022-03-28

Introdution

We scraped all of the data used in the following analysis from baseballsavant.com, the online depot for all publicly available advanced metrics on Major League Baseball players. Our sample is 131 of the 132 MLB hitters who "qualified" for end-of-year awards during the 2021 season by recording at least 502 plate appearances. Our data set is a collection of these players' 2021 statistics.

We chose to remove one particular observation because our previous analysis of this data set revealed that this player is a significant outlier when it comes to average launch angle (LA), one of our predictor variables of interest. Launch angle is the vertical angle at which the ball leaves a hitter's bat, where 0° is parallel to the ground and 90° is straight up in the air, and the outlier had a negative launch angle which implied that player often batted balls straight into the ground. Thus, to maintain our population as MLB qualifiers, we removed this player who clearly performed badly. Our concern was that this player was such an outlier that the ISO vs. LA model we built was entirely unable to predict him correctly. The population we will generalize our findings to is all MLB players with similar plate appearance quantities and positive LA, regardless of year.

The response variable we are interested in modeling/predicting is Isolated Power (ISO). Isolated power is meant to quantify how much power a hitter demonstrates during games. It calculates the rate at which they hit for "extra" total bases. Where a single is 1 total base, a double is 2, a triple is 3, and a home run is 4, ISO is calculated by $ISO = \frac{\text{(Total Bases) - (Singles)}}{\text{(At Bats)}}$. Our goal in this analysis is to use statistics that measure innate player skills and/or tendencies to predict how much power a player will demonstrate during games (according to ISO) with the highest possible accuracy.

Predictor variables under consideration

The first of our predictor variables of interest is **barrel percentage (BRL%)**. BRL% combines information about the launch angles and exit velocities of a hitter's batted balls. In our previous analysis of this data set, we found that 31.5% of the variation in ISO could be explained by average launch angle alone. Exit velocity is the velocity of the ball in MPH immediately after being hit in play by the batter. If a batter hits the ball with a 98 MPH exit velocity, it must be hit between 26 and 30° to be defined as a barrel. If they hit it at 99 MPH, it must be hit between 25 and 31°. This pattern continues–every 1 MPH increase in exit velocity loosens the launch angle requirement by 2 total degrees (one in each direction). We are also interested in the somewhat aforementioned **average launch angle (LA)**, **average exit velocity (EV)**, and two statistics that provide almost identical information (**sweet spot percentage** and **hard hit rate**). These variables will obviously correlate very highly with BRL%, but we want to see which correlate best with our response variable in the bivariate sense.

Another predictor of interest is **pull percentage (PULL%)**, or the percentage of a hitter's batted balls that are hit to the same third of the field as the side of home plate from which they bat (e.g. right vs. left).

Finally, we are also interested in including swing discipline and contact ability statistics as predictors in our model. We anticipate significant correlation between at least a few of these such variables. Not all will be simultaneously used in our final model. The variables are:

Zone swing percentage (Z-Swing%), the percentage of pitches in the strike zone that the batter swings at.

Zone swing and miss percentage (Z-Miss%), the percentage of pitches in the strike zone that the batter swings at and misses.

Out of zone swing percentage (OZ-Swing%), the percentage of pitches in outside of the strike zone that the batter swings at.

Out of zone swing and miss percentage (OZ-Miss%), the percentage of pitches outside of the strike zone that the batter swings at and misses.

Walk rate (BB%), the percentage of plate appearances in which the batter draws a walk (by not swinging at 4 out-of-zone pitches).

Each of the other variables in our model were not of interest for a variety of reasons. Chief among them: ISO was dependent upon the variable in an objective mathematical sense (ex. batting_avg), the variable was already sufficiently represented by others in the model that were more interesting (ex. opposite_percent), or the variable was a cumulative statistic. In our previous analysis of this data set we decided we would not use cumulative stats because of the differences in plate appearance totals from player to player.

Checking for multicollinearity

As we can see in Figure 1, some correlation exists between all variables as expected. Our subjective threshold for where correlation between predictor variables becomes too significant to include both in the model is at strength 0.6. Of all the predictor variables, BRL% had the highest correlation with ISO at 0.877. We decided to remove the other variables that gave information about launch angle or exit velocity (LA, EV, SwSp%, HH%) out of the concern that they correlated too highly with BRL% or were otherwise redundant. These plots confirm that at least EV (0.681 correlation) and HH% (0.682 correlation) seemed to be highly correlated with BRL%.

PULL% had a .575 correlation with ISO and non-significant correlation with the other variables we ended up including.

BB% had a .361 correlation with ISO and non-significant correlation with the other variables we ended up including.

Among the other variables of interest, only Z-Swing% was uncorrelated with any of our other predictor variables of interest. It had a correlation of strength .368 with ISO. For an example of a variable that was rejected, see Z-Miss%. It had a correlation of strength .703 with BRL%, raising concerns about multicollinearity, hence we will take Z-Miss% out of consideration for the remainder of this analysis.

After performing this multicollinearity test, we were left with four variables of interest: BRL%, PULL%, BB%, and Z-Swing%. The strengths of correlation between BRL%:PULL% (0.431) and BRL%:BB% (0.419) remain somewhat high. We will focus on these relationships when checking to see if interaction variables are necessary.

Feature engineering

In the following Figures 2, 3, 4, 5, let's take a closer look at the bivariate relationships between each predictor and our response variable to see if any feature engineering is required.

These plots indicate that the LINE conditions (linearity, independence of data points, normality of errors, and equal variance/homoskedasticity) for each predictor-response relationship are strong.

If we have one concern, it is in the last set of plots (ISO vs. BB%). From this last residual plot, the LINE conditions appear to hold as well, but there is one player that is a pretty massive outlier in terms of BB% (Juan Soto). His ISO doesn't appear to be much more than 1 standard deviation away from the value predicted by our model, though, indicating that this data point does not radically alter our model. Conclude that feature engineering is not necessary.

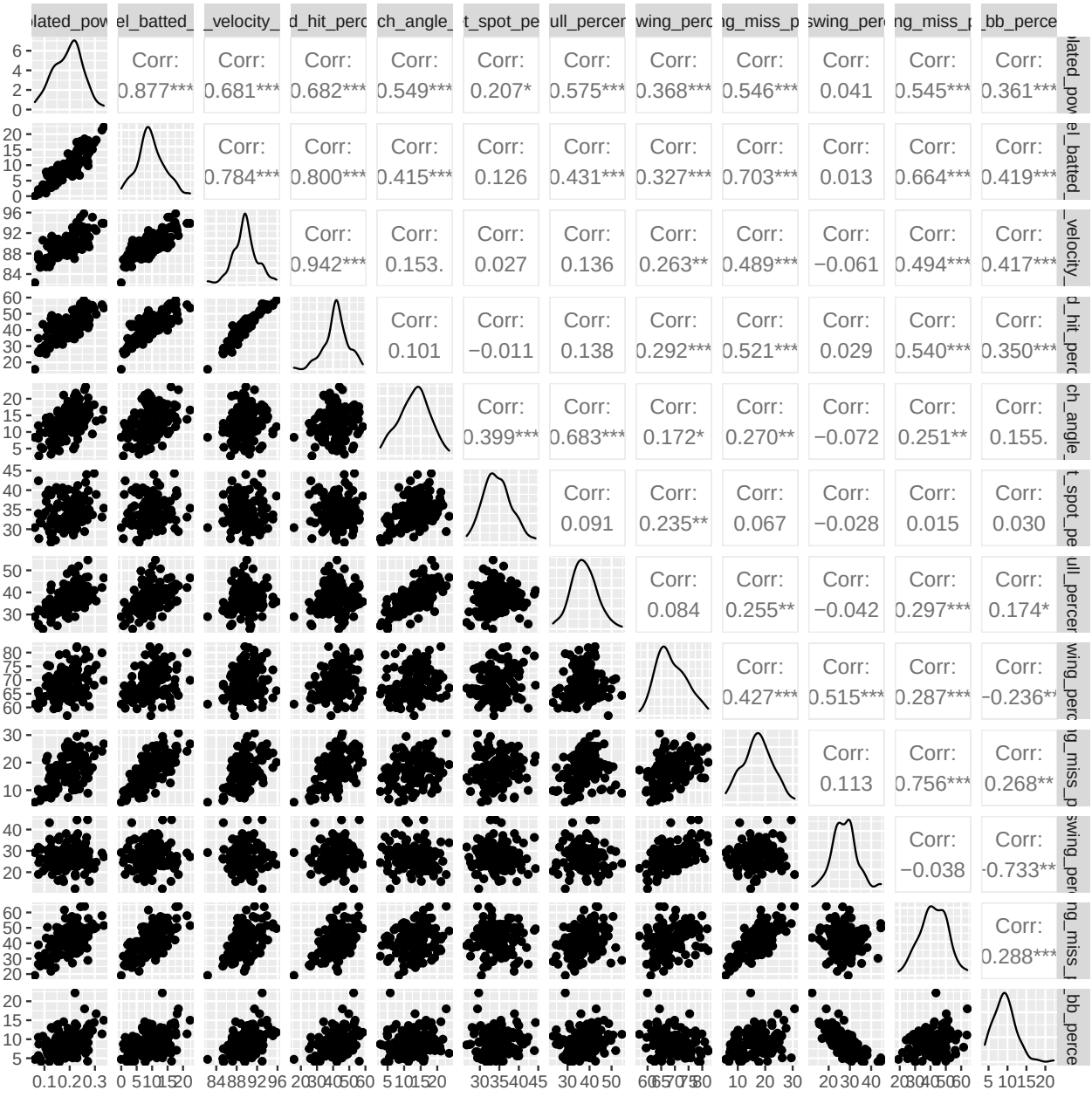

Figure 1: Correlation between predictor variables

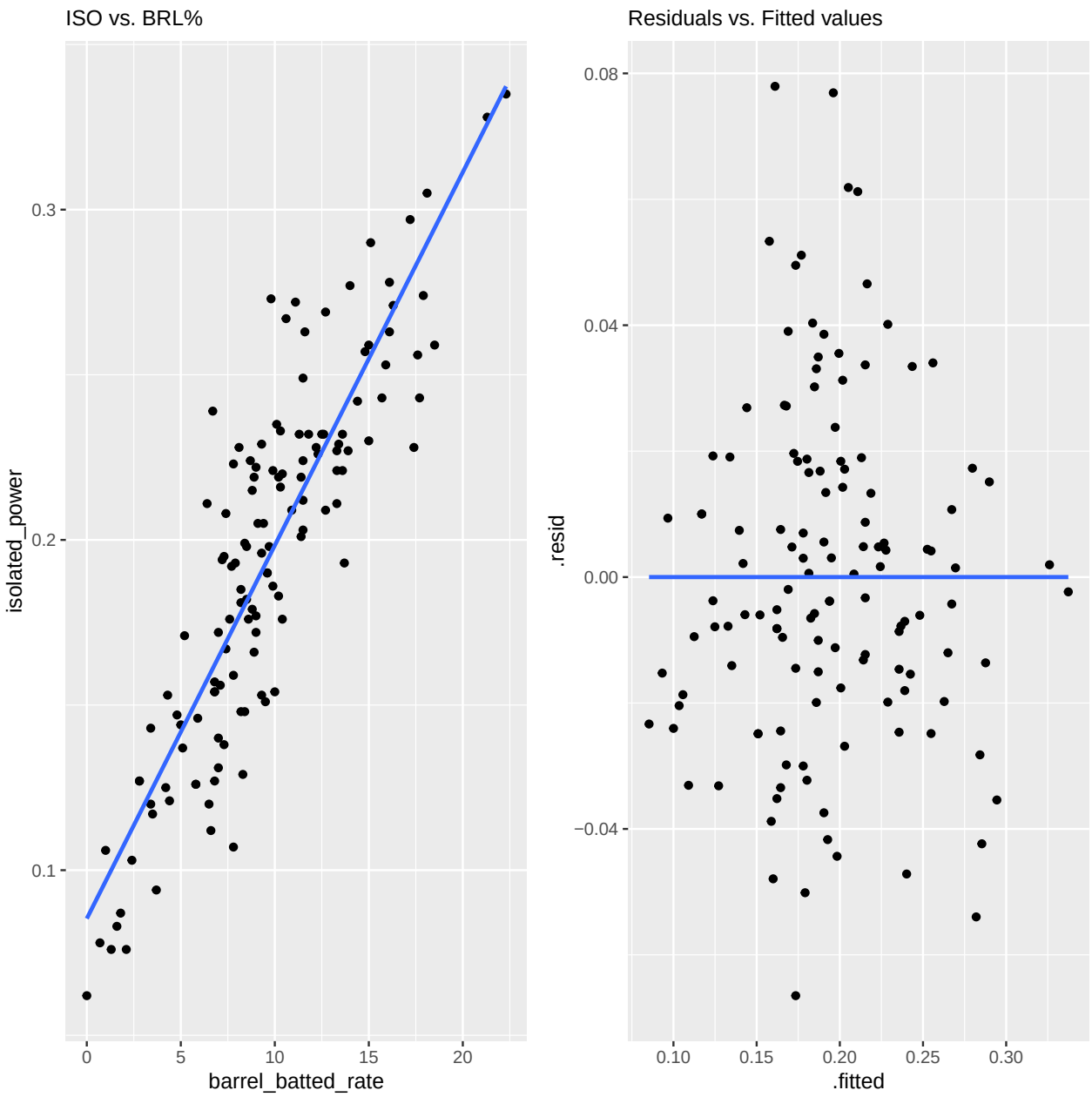

Figure 2: Isolated power and Barrel Batted Rate

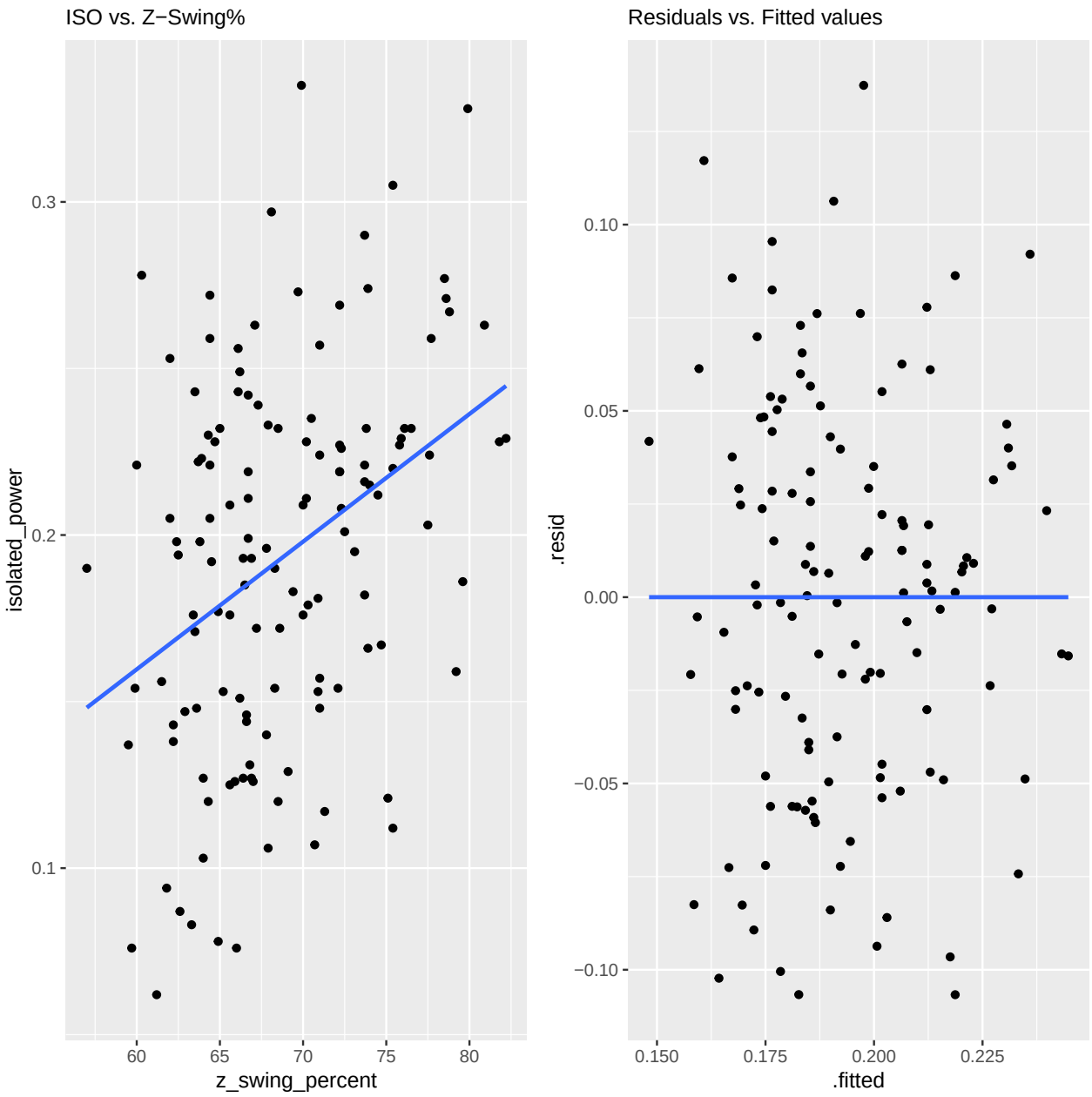

Figure 3: Isolated Power and Zone Swing Percentage

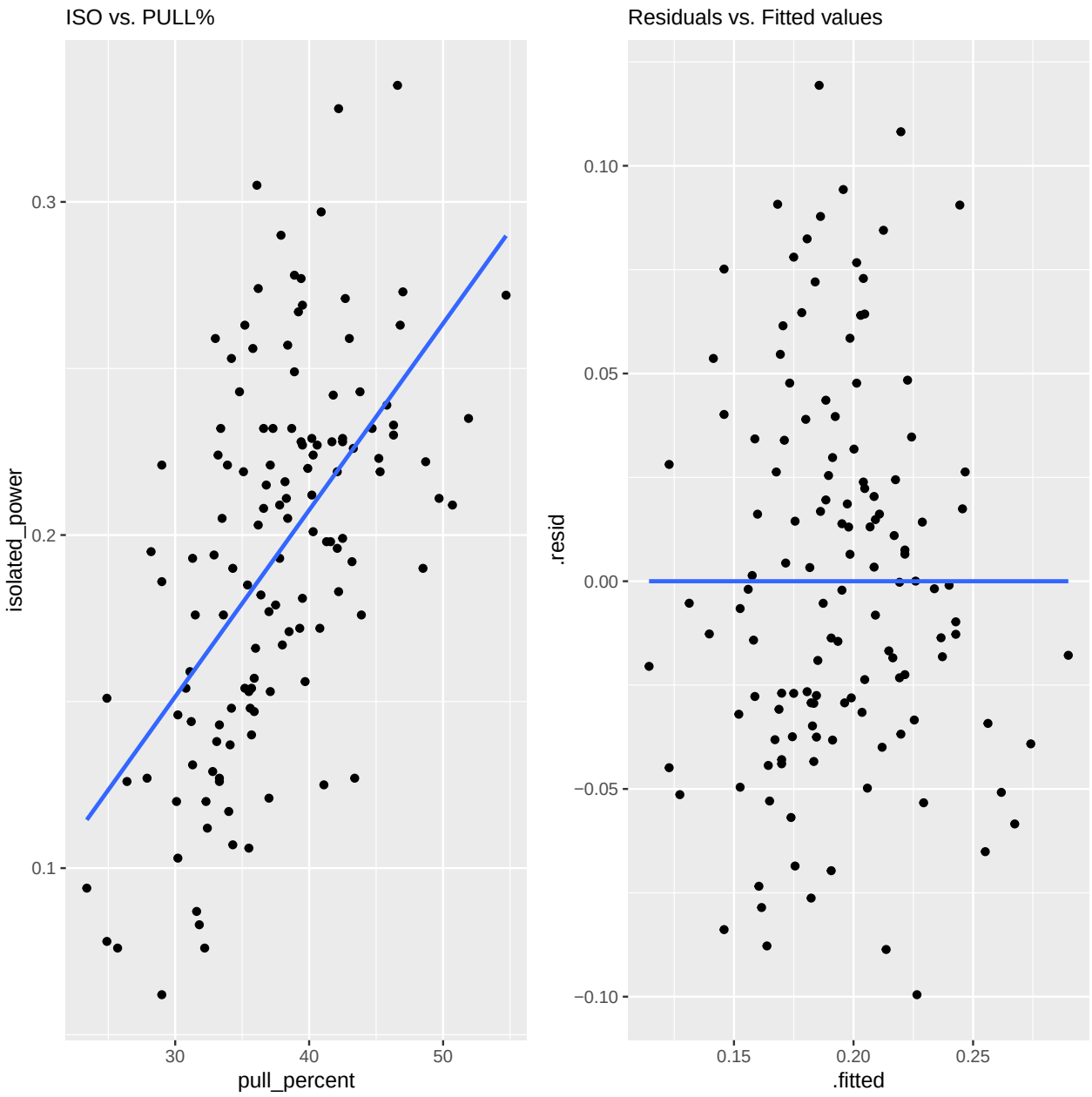

Figure 4: Isolated Power and Pull Percentage

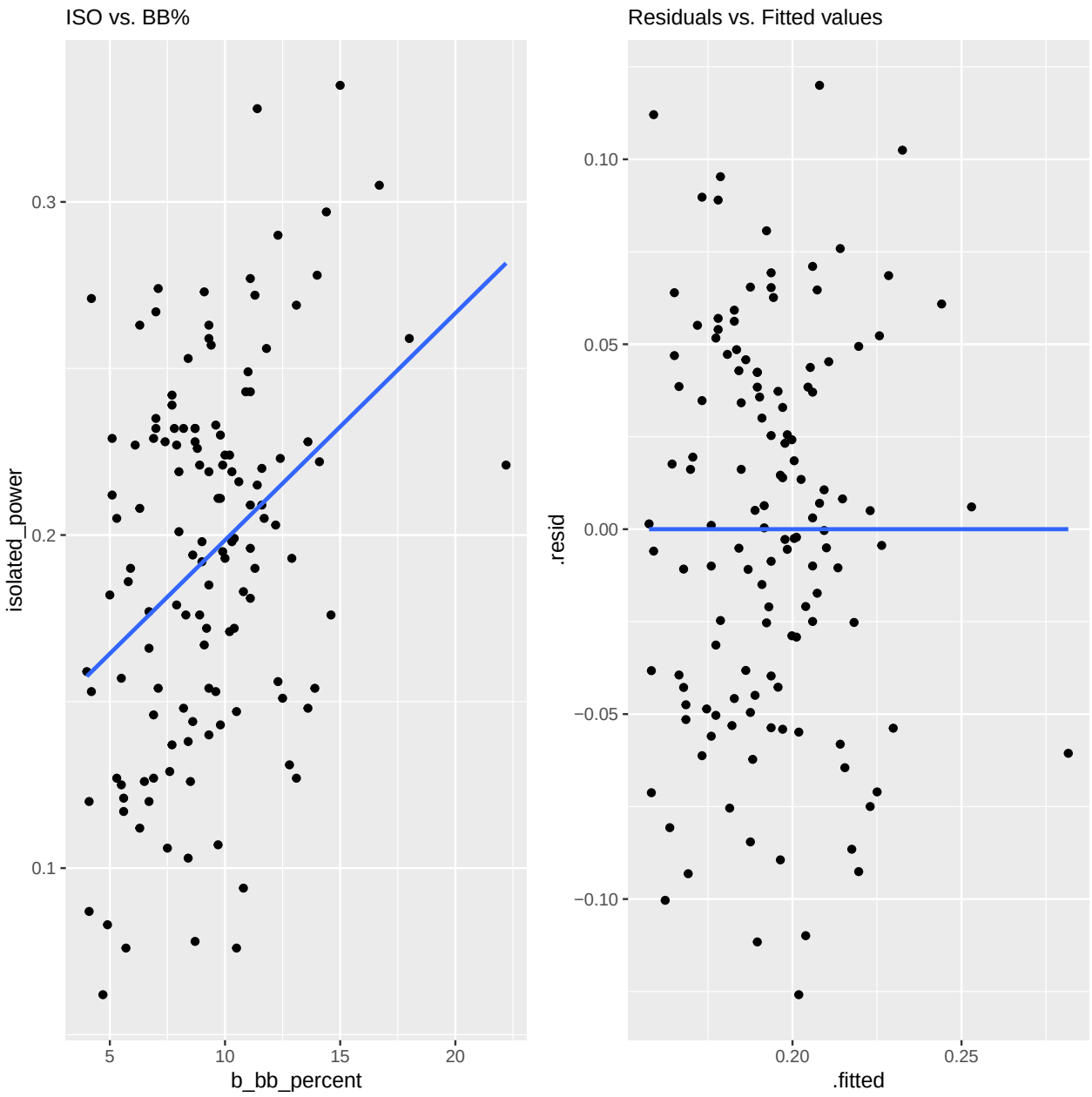

Figure 5: Isolated Power and Walk Rate

Interaction variables

```
## # A tibble: 11 x 5
##    term                                 estimate std.error statistic p.value
##    <chr>                                   <dbl>     <dbl>     <dbl>   <dbl>
##  1 (Intercept)                          -0.0987    0.253       -0.390   0.697
##  2 barrel_batted_rate                    0.00443   0.00810      0.547   0.585
##  3 z_swing_percent                      -0.000354  0.00355     -0.0996  0.921
##  4 pull_percent                          0.00910   0.00724      1.26    0.211
##  5 b_bb_percent                         -0.00250   0.00921     -0.272   0.786
##  6 barrel_batted_rate:z_swing_percent    0.000173  0.000110     1.56    0.121
##  7 barrel_batted_rate:pull_percent      -0.000214  0.000109    -1.95    0.0532
##  8 barrel_batted_rate:b_bb_percent       0.0000977 0.000155     0.631   0.529
##  9 z_swing_percent:pull_percent         -0.0000419 0.0000978   -0.428   0.669
## 10 z_swing_percent:b_bb_percent          0.000134  0.000132     1.02    0.310
## 11 pull_percent:b_bb_percent            -0.000187  0.000164    -1.14    0.255
```

In our multicollinearity analysis, we noted that BRL% had a non-significant (according to our own definition) yet slightly concerning amount of correlation with PULL% and BB%. The p-values here corroborate this. None of the bivariate relationships yielded p-values below alpha=0.05, so we did not reject the null hypothesis (H_0 = no relationship) in any of the above cases. We did come very close to rejecting the null when it came to the BRL%:PULL% relationship, however (p=0.0532). Just in case, we will run a nested F test to see if both variables are necessary in the model.

```
## Analysis of Variance Table
##
## Response: isolated_power
##                     Df   Sum Sq  Mean Sq F value    Pr(>F)
## z_swing_percent      1 0.056313 0.056313  100.11 < 2.2e-16 ***
## b_bb_percent         1 0.088123 0.088123  156.66 < 2.2e-16 ***
## pull_percent         1 0.089399 0.089399  158.92 < 2.2e-16 ***
## barrel_batted_rate   1 0.111103 0.111103  197.51 < 2.2e-16 ***
## Residuals          126 0.070878 0.000563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Analysis of Variance Table
##
## Response: isolated_power
##                                      Df   Sum Sq  Mean Sq F value    Pr(>F)
## z_swing_percent                       1 0.056313 0.056313  69.582 1.048e-13
## b_bb_percent                          1 0.088123 0.088123 108.888 < 2.2e-16
## c(pull_percent + barrel_batted_rate)  1 0.168598 0.168598 208.326 < 2.2e-16
## Residuals                           127 0.102781 0.000809
##
## z_swing_percent                      ***
## b_bb_percent                         ***
## c(pull_percent + barrel_batted_rate) ***
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Analysis of Variance Table
##
## Model 1: isolated_power ~ z_swing_percent + b_bb_percent + pull_percent +
##     barrel_batted_rate
```

```
## Model 2: isolated_power ~ z_swing_percent + b_bb_percent + c(pull_percent +
##     barrel_batted_rate)
##   Res.Df      RSS Df Sum of Sq      F  Pr(>F)
## 1    126 0.070878
## 2    127 0.102781 -1 -0.031903 56.714 8.5e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Given a P value of 8.5e-12, we reject the null hypothesis that BRL% and PULL% share the same coefficient. Interaction variables are not necessary.

Comparing two models using cross validation

Although we concluded that an interaction variable of BRL% and PULL% was not necessary in the previous section, it was a borderline decision, hence hypothesize that having both of these variables may reduce the fit of the model, and prioritize BRL% since it correlated highly with our response variable compared to PULL%.

Define a full model with explanatory variables BRL%, PULL%, BB%, and Z-Swing%, and a reduced model that has all of these but PULL%, and compare the two through 3-fold cross validation with training data which is two-thirds of our dataset.

Our full model:

```
## # A tibble: 5 x 5
##   term               estimate std.error statistic  p.value
##   <chr>                 <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)        -0.0854   0.0467       -1.83 7.11e- 2
## 2 z_swing_percent     0.00154  0.000562      2.74 7.57e- 3
## 3 b_bb_percent        0.00115  0.00105       1.10 2.77e- 1
## 4 pull_percent        0.00202  0.000487      4.15 8.11e- 5
## 5 barrel_batted_rate  0.00912  0.000814     11.2  3.39e-18
```

Our reduced model:

```
## # A tibble: 4 x 5
##   term               estimate std.error statistic  p.value
##   <chr>                 <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)        0.00763   0.0448       0.170 8.65e- 1
## 2 z_swing_percent    0.00117   0.000607     1.92  5.78e- 2
## 3 b_bb_percent       0.000407  0.00113      0.360 7.20e- 1
## 4 barrel_batted_rate 0.0107    0.000788    13.6   8.75e-23
```

```
collect_metrics(fit_rs_full)
```

```
## # A tibble: 2 x 6
##   .metric .estimator   mean     n std_err .config
##   <chr>   <chr>       <dbl> <int>   <dbl> <chr>
## 1 rmse    standard   0.0239     3 0.00150 Preprocessor1_Model1
## 2 rsq     standard   0.778      3 0.105   Preprocessor1_Model1
```

```
collect_metrics(fit_rs_reduced)
```

```
## # A tibble: 2 x 6
##   .metric .estimator   mean     n  std_err .config
##   <chr>   <chr>       <dbl> <int>    <dbl> <chr>
## 1 rmse    standard   0.0259     3 0.000309 Preprocessor1_Model1
## 2 rsq     standard   0.756      3 0.0856   Preprocessor1_Model1
```

Below are the R^2 and RMSE of the full model on the test data:

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.811
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard      0.0246
```

Below are the R^2 and RMSE of the reduced model on the test data:

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.729
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard      0.0292
```

Since our full model has a higher R^2 and a lower RMSE on both the training and the test data, we will choose the full model over the reduced model out of the two.

Choosing a statistical model

Now, instead of choosing predictor variables ourselves, run the forward selection algorithm (forward and backward selection yield the same model) on all predictor variables listed initially, with the selection criteria being the F-statistic.

```
#FULL mod
stats::step(lm(isolated_power ~ 1, data=Batters_train),
            isolated_power ~ exit_velocity_avg + launch_angle_avg + sweet_spot_percent + hard_hit_percen
            direction = "forward", test = "F")
```

```
## Start:  AIC=-498.07
## isolated_power ~ 1
##
##                          Df Sum of Sq      RSS     AIC  F value    Pr(>F)
## + barrel_batted_rate      1  0.219791 0.057689 -632.72 323.8438 < 2.2e-16 ***
## + hard_hit_percent        1  0.124967 0.152513 -548.14  69.6479 1.139e-12 ***
## + exit_velocity_avg       1  0.118821 0.158659 -544.70  63.6572 6.253e-12 ***
## + z_swing_miss_percent    1  0.105590 0.171890 -537.73  52.2143 1.990e-10 ***
## + launch_angle_avg        1  0.096722 0.180757 -533.36  45.4831 1.759e-09 ***
## + oz_swing_miss_percent   1  0.087741 0.189739 -529.14  39.3066 1.446e-08 ***
## + pull_percent            1  0.086600 0.190879 -528.62  38.5639 1.876e-08 ***
## + z_swing_percent         1  0.035777 0.241703 -508.08  12.5816 0.0006368 ***
## + b_bb_percent            1  0.027095 0.250384 -505.01   9.1983 0.0032119 **
## + sweet_spot_percent      1  0.018683 0.258796 -502.13   6.1363 0.0152236 *
## <none>                               0.277479 -498.07
## + oz_swing_percent        1  0.000470 0.277009 -496.22   0.1443 0.7049705
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=-632.72
## isolated_power ~ barrel_batted_rate
```

```
##
##                         Df Sum of Sq      RSS     AIC F value    Pr(>F)
## + launch_angle_avg       1 0.0116252 0.046064 -650.30 21.1993 1.457e-05 ***
## + pull_percent           1 0.0081182 0.049571 -643.91 13.7567 0.0003725 ***
## + sweet_spot_percent     1 0.0052672 0.052422 -639.05  8.4402 0.0046901 **
## + z_swing_percent        1 0.0028711 0.054818 -635.16  4.3995 0.0389543 *
## + z_swing_miss_percent   1 0.0013359 0.056353 -632.76  1.9914 0.1618905
## <none>                               0.057689 -632.72
## + oz_swing_miss_percent  1 0.0010798 0.056609 -632.36  1.6022 0.2090881
## + exit_velocity_avg      1 0.0010422 0.056647 -632.30  1.5455 0.2172645
## + hard_hit_percent       1 0.0008428 0.056846 -632.00  1.2453 0.2676281
## + b_bb_percent           1 0.0005163 0.057173 -631.50  0.7585 0.3862679
## + oz_swing_percent       1 0.0002823 0.057407 -631.14  0.4131 0.5221296
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=-650.3
## isolated_power ~ barrel_batted_rate + launch_angle_avg
##
##                         Df  Sum of Sq      RSS     AIC F value  Pr(>F)
## + z_swing_percent        1 0.00202938 0.044034 -652.22  3.8252 0.05385 .
## + sweet_spot_percent     1 0.00126233 0.044801 -650.71  2.3386 0.13000
## + z_swing_miss_percent   1 0.00122467 0.044839 -650.64  2.2669 0.13596
## + pull_percent           1 0.00120032 0.044863 -650.59  2.2207 0.13996
## <none>                               0.046064 -650.30
## + oz_swing_miss_percent  1 0.00049070 0.045573 -649.23  0.8937 0.34723
## + hard_hit_percent       1 0.00044911 0.045615 -649.15  0.8172 0.36862
## + oz_swing_percent       1 0.00015216 0.045912 -648.58  0.2751 0.60134
## + b_bb_percent           1 0.00006362 0.046000 -648.42  0.1148 0.73560
## + exit_velocity_avg      1 0.00004084 0.046023 -648.37  0.0736 0.78677
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=-652.22
## isolated_power ~ barrel_batted_rate + launch_angle_avg + z_swing_percent
##
##                         Df  Sum of Sq      RSS     AIC F value  Pr(>F)
## + z_swing_miss_percent   1 0.00271094 0.041323 -655.74  5.3794 0.02286 *
## + pull_percent           1 0.00175052 0.042284 -653.75  3.3947 0.06902 .
## <none>                               0.044034 -652.22
## + sweet_spot_percent     1 0.00069980 0.043335 -651.61  1.3242 0.25318
## + oz_swing_miss_percent  1 0.00059285 0.043441 -651.40  1.1191 0.29323
## + oz_swing_percent       1 0.00048299 0.043551 -651.18  0.9094 0.34308
## + b_bb_percent           1 0.00044014 0.043594 -651.09  0.8279 0.36555
## + hard_hit_percent       1 0.00022708 0.043807 -650.67  0.4251 0.51625
## + exit_velocity_avg      1 0.00000420 0.044030 -650.22  0.0078 0.92971
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=-655.74
## isolated_power ~ barrel_batted_rate + launch_angle_avg + z_swing_percent +
##     z_swing_miss_percent
##
##                         Df  Sum of Sq      RSS     AIC F value  Pr(>F)
```

```
## + pull_percent           1 0.00183011 0.039493 -657.69  3.7535 0.05618 .
## <none>                               0.041323 -655.74
## + b_bb_percent           1 0.00060881 0.040715 -655.04  1.2112 0.27436
## + sweet_spot_percent     1 0.00054991 0.040773 -654.91  1.0924 0.29904
## + oz_swing_percent       1 0.00033811 0.040985 -654.46  0.6682 0.41607
## + hard_hit_percent       1 0.00013280 0.041191 -654.02  0.2612 0.61072
## + exit_velocity_avg      1 0.00000909 0.041314 -653.76  0.0178 0.89415
## + oz_swing_miss_percent  1 0.00000005 0.041323 -653.74  0.0001 0.99179
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=-657.69
## isolated_power ~ barrel_batted_rate + launch_angle_avg + z_swing_percent +
##     z_swing_miss_percent + pull_percent
##
##                         Df  Sum of Sq      RSS     AIC F value Pr(>F)
## + sweet_spot_percent     1 0.00111056 0.038383 -658.17  2.3147 0.1321
## + b_bb_percent           1 0.00093613 0.038557 -657.77  1.9423 0.1673
## <none>                               0.039493 -657.69
## + oz_swing_percent       1 0.00055133 0.038942 -656.91  1.1326 0.2904
## + hard_hit_percent       1 0.00054071 0.038953 -656.88  1.1105 0.2951
## + exit_velocity_avg      1 0.00016019 0.039333 -656.04  0.3258 0.5697
## + oz_swing_miss_percent  1 0.00003954 0.039454 -655.77  0.0802 0.7778
##
## Step:  AIC=-658.17
## isolated_power ~ barrel_batted_rate + launch_angle_avg + z_swing_percent +
##     z_swing_miss_percent + pull_percent + sweet_spot_percent
##
##                         Df  Sum of Sq      RSS     AIC F value Pr(>F)
## + b_bb_percent           1 0.00091706 0.037466 -658.27  1.9337 0.1683
## <none>                               0.038383 -658.17
## + hard_hit_percent       1 0.00082217 0.037561 -658.05  1.7292 0.1923
## + oz_swing_percent       1 0.00051019 0.037873 -657.33  1.0642 0.3054
## + exit_velocity_avg      1 0.00035937 0.038023 -656.99  0.7466 0.3902
## + oz_swing_miss_percent  1 0.00002678 0.038356 -656.23  0.0551 0.8149
##
## Step:  AIC=-658.27
## isolated_power ~ barrel_batted_rate + launch_angle_avg + z_swing_percent +
##     z_swing_miss_percent + pull_percent + sweet_spot_percent +
##     b_bb_percent
##
##                         Df  Sum of Sq      RSS     AIC F value Pr(>F)
## <none>                               0.037466 -658.27
## + hard_hit_percent       1 0.00061997 0.036846 -657.72  1.3124 0.2555
## + exit_velocity_avg      1 0.00012405 0.037342 -656.56  0.2591 0.6122
## + oz_swing_miss_percent  1 0.00002987 0.037436 -656.34  0.0622 0.8036
## + oz_swing_percent       1 0.00000408 0.037462 -656.28  0.0085 0.9268
##
## Call:
## lm(formula = isolated_power ~ barrel_batted_rate + launch_angle_avg +
##     z_swing_percent + z_swing_miss_percent + pull_percent + sweet_spot_percent +
##     b_bb_percent, data = Batters_train)
##
```

```
## Coefficients:
##          (Intercept)    barrel_batted_rate      launch_angle_avg
##            -0.105701              0.010133              0.001477
##      z_swing_percent  z_swing_miss_percent          pull_percent
##              0.001650             -0.001788              0.001358
##   sweet_spot_percent          b_bb_percent
##              0.001095              0.001359
```

The algorithm suggests a 7-variable model of barrel_batted_rate, launch_angle_avg, z_swing_percent, z_swing_miss_percent, pull_percent, sweet_spot_percent, and b_bb_percent.

Final statistics and analysis

We decide to choose the simpler computational model since the statistical model has many instances of multicollinearity. We test the significance of the model's coefficients through overall F tests, with the null hypotheses that each $\beta_i = 0$ and alternative hypotheses that $\beta_i \neq 0$.

```
final_mod <- lm(isolated_power ~ z_swing_percent + b_bb_percent + pull_percent + barrel_batted_rate,
  data = Batters)

final_mod %>% anova()
```

```
## Analysis of Variance Table
##
## Response: isolated_power
##                      Df   Sum Sq  Mean Sq F value    Pr(>F)
## z_swing_percent       1 0.056313 0.056313  100.11 < 2.2e-16 ***
## b_bb_percent          1 0.088123 0.088123  156.66 < 2.2e-16 ***
## pull_percent          1 0.089399 0.089399  158.92 < 2.2e-16 ***
## barrel_batted_rate    1 0.111103 0.111103  197.51 < 2.2e-16 ***
## Residuals           126 0.070878 0.000563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since all of the p-values are very small, reject each null hypothesis and conclude that all variables are significant.

From the section "Comparing two models using cross validation," we had found that our final model (the full model), fit to the testing data, had $R^2 = 0.811$ and $RMSE = 0.246$ which is quite ideal. Nevertheless, while a high R^2 means the model can explain a large proportion of the variability, it does not necessarily guarantee an accurate description of the population since the model may be overfitting and explaining random variability as part of the model. However, our process of cross-validation and comparing train and test data results diminish this argument and convincingly demonstrate that this model can be highly representative of the population.

Another reason a high R^2 is not necessarily good is that outliers with high leverage and influence may deceivingly produce a high R^2 when the model, considered without these outliers, is not at all an accurate description of our model. However, from the following residual plot, we can see that no particular observations have particularly high leverage or influence.

```
final_mod %>%
  augment() %>%
  ggplot(aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_smooth(formula = y ~ x, method = "lm", se = FALSE)
```

Lastly, let us draw some predictions from our model.

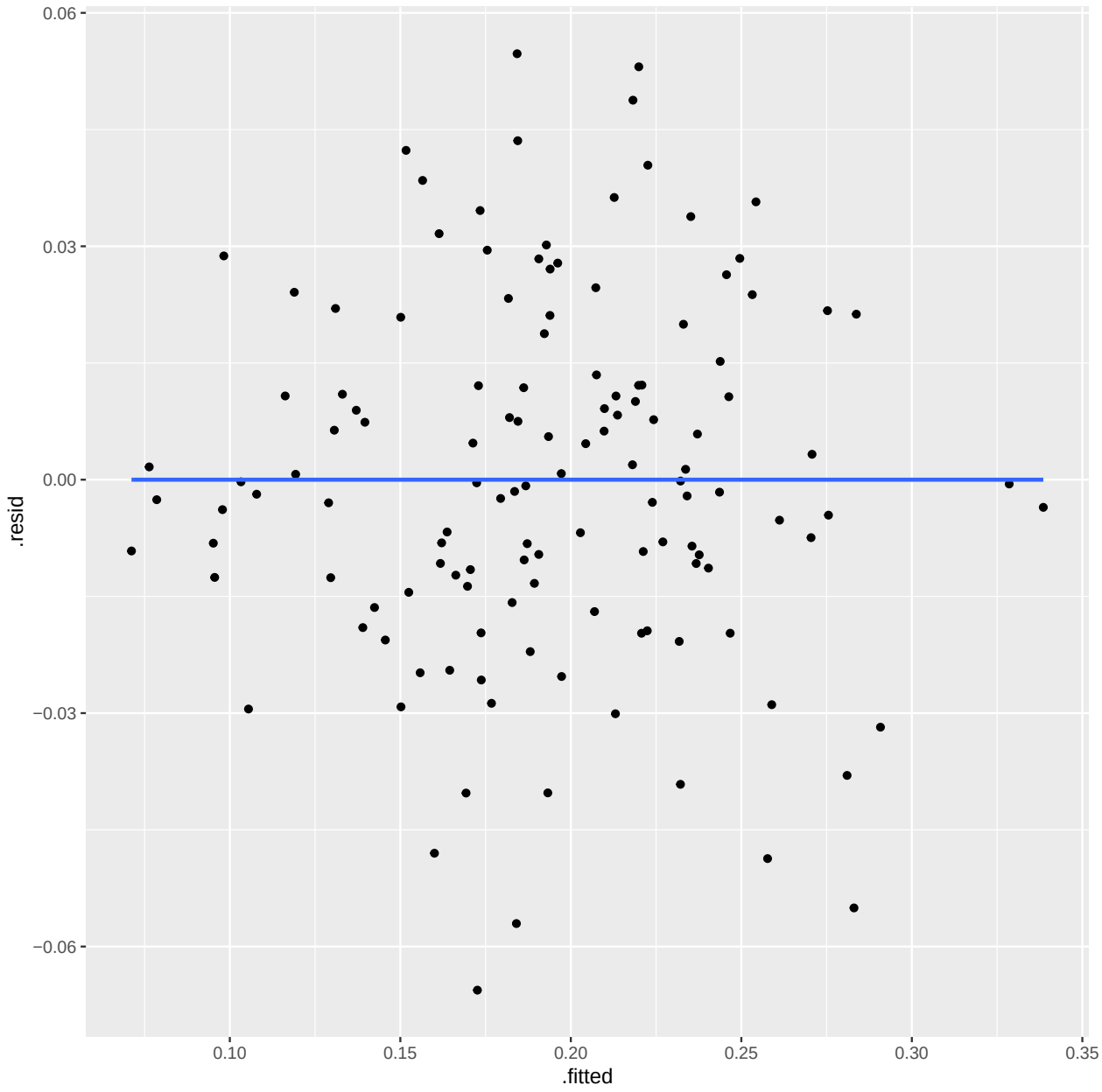

Figure 6: Residual Plot of Final Model

```
## # A tibble: 1 x 7
##   z_swing_percent b_bb_percent pull_percent barrel_batted_rate .fitted .lower
##             <dbl>        <dbl>        <dbl>              <dbl>   <dbl>  <dbl>
## 1            69.9           15         46.6               22.3   0.339  0.325
## # ... with 1 more variable: .upper <dbl>
```

```
## # A tibble: 1 x 7
##   z_swing_percent b_bb_percent pull_percent barrel_batted_rate .fitted .lower
##             <dbl>        <dbl>        <dbl>              <dbl>   <dbl>  <dbl>
## 1            69.9           15         46.6               22.3   0.339  0.290
## # ... with 1 more variable: .upper <dbl>
```

Taking Shohei Ohtani's metrics as our explanatory variables, since he had the highest isolated power percentage of 0.335 in this dataset, our model gives a 95% confidence interval of [0.325, 0.352] and a 95% prediction interval of [0.290, 0.387] for the isolated power percentage (rounded to three significant figures).

Summary

Our intention with this analysis was to create a model that best predicts the ISOs of qualified MLB players using statistics that measure innate player skills and/or tendencies. We began by setting our focus on a large group of such predictors, and narrowed that focus by testing for multicollinearity. The four variables we were interested in building our model with were BRL%, PULL%, BB%, and Z-Swing%. After confirming that feature engineering and interaction variables were not necessary, we aimed to find the combination of these variables that, when placed into a model, yielded the strongest predictions. A simple computational model containing BRL%, PULL%, BB%, and Z-Swing% ended up being the best model when accounting for multicollinearity. This model had $R^2 = 0.811$, a great improvement over our model regressing ISO on LA in the SLR analysis, which produced $R^2 = 0.315$.