# Final Project

Benji Gold and Sam Alkalay

## Introduction

Baseball, America's pastime, has a long and storied tradition that dates back well over 100 years. Since the 1850's, some form of statistics measuring how good a player is has been tracked. This began through the use of the box score, which tracked basic statistics, such as hits, runs, and errors, from which a player's batting average can be constructed. Over one hundred years later, a pioneering statistician by the name of Bill James introduced new statistical concepts, such as on-base percentage and runs created, in his annual Baseball Abstract (Lee 2018). As technology has improved, the statistics being tracked became more and more sophisticated. Then, in 2015 analytics in baseball took a giant leap. With the introduction of Statcast, teams were able to track novel metrics, such as a batter's exit velocity (the speed of the baseball as it comes off the bat, immediately after a batter makes contact) and barrel percentage (the percentage of baseballs hit off of the player's barrel) ("Statcast Search"). Around the league, teams adopted these new statistics to try and gain a competitive advantage, through which they would be able to better predict a player's potential. However, is this actually the case? While these new statistics are widely used, it is unclear whether they actually provide any useful information for predicting a player's potential. This research project intends to explore that idea through the use of a logistic regression model to predict whether a player is an all-star. The research question of interest is:

Do old or new wave statistics do a better job at predicting whether a player is selected as an all-star?

The response variables of interest are: All.Star: Whether a player is selected as an all-star. Salary: How much money a player makes.

For our analysis, we have selected two datasets. The first is from Baseball Reference, which consists of standard statistics that offer a broad view of a player's performance in a particular season. The second is from Statcast, which consists of each player's primary position. ADD MORE ABOUT WHAT WE DID WITH THE DATA HERE

## Methodology

## Results

## Discussion

## Packages and Data

```
library(tidyverse)
library(tidymodels)
library(glmnet)
library(caret)
library(MASS)
library(lme4)
stats <- read.csv("data/stats.csv")


stats <- replace(stats, stats =="", NA)
stats <- stats %>%
  drop_na() %>%
  mutate(AVG300 = case_when(batting_avg >= .3 ~ "Greater than 300", TRUE ~ "Less than 300"
         HR40 = case_when(b_home_run >= 40 ~ "Greater than 40", TRUE ~ "Less than 40"), pi
view(stats)
```

## Lassos for Variable Selection

```
# LASSO Variable Selection Basic Stats
y <- stats$All.Star
x <- model.matrix(All.Star ~ player_age + b_ab + b_total_pa + b_total_hits + b_home_run +
                    b_double + b_triple + b_home_run * HR40 + b_strikeout + b_walk +
                    batting_avg + slg_percent + on_base_percent + Position, data = stats)
m_lasso_cv <- cv.glmnet(x, y, alpha = 1)
best_lambda <- m_lasso_cv$lambda.min
best_lambda
```

```
[1] 0.000866343
```

```
m_best <- glmnet(x, y, alpha = 1, lambda = best_lambda)
m_best$beta
```

```
29 x 1 sparse Matrix of class "dgCMatrix"
                                              s0
(Intercept)                         .
player_age                         -0.0029980849
b_ab                               -0.0014075333
b_total_pa                          .
b_total_hits                        0.0036214759
b_home_run                          0.0121683578
AVG300Less than 300                -0.1320213561
batting_avg                         .
b_double                            0.0026194103
b_triple                            0.0046847465
HR40Less than 40                   -0.1039443676
b_strikeout                        -0.0008468996
b_walk                              0.0030687942
slg_percent                         0.0404013686
on_base_percent                    -0.2224740295
Position2B                          0.0345979059
Position3B                         -0.0366558483
PositionC                           0.0605302619
PositionCF                          0.0493821587
PositionCH                          .
PositionDH                         -0.0385492210
PositionDNP                        -0.0063018573
PositionLF                         -0.0524369335
PositionPH                          .
PositionRF                          .
PositionSP                          0.1745066736
PositionSS                          0.0427289127
AVG300Less than 300:batting_avg     .
b_home_run:HR40Less than 40         .
```

```r
# LASSO Variable Selection Advanced Stats
y <- stats$All.Star
x <- model.matrix(All.Star ~ player_age + launch_angle_avg + sweet_spot_percent +
                    barrel + solidcontact_percent + flareburner_percent +
                    hard_hit_percent + avg_hyper_speed + z_swing_percent +
                    oz_swing_percent + meatball_swing_percent, data = stats)
m_lasso_cv <- cv.glmnet(x, y, alpha = 1)
best_lambda <- m_lasso_cv$lambda.min
best_lambda
```

```
[1] 0.00530318
```

```
m_best <- glmnet(x, y, alpha = 1, lambda = best_lambda)
m_best$beta
```

```
12 x 1 sparse Matrix of class "dgCMatrix"
                                s0
(Intercept)             .
player_age             -0.0019142646
launch_angle_avg       -0.0001873012
sweet_spot_percent      .
barrel                  0.0081373961
solidcontact_percent   -0.0031776615
flareburner_percent    -0.0018772377
hard_hit_percent       -0.0019055476
avg_hyper_speed         .
z_swing_percent         .
oz_swing_percent        .
meatball_swing_percent -0.0022336733
```

## Regressions

```
#Basic model
m1 <- glm(All.Star ~ player_age + b_ab +  b_total_hits +
                    b_double + b_triple + b_home_run + b_strikeout +
                    b_bb_percent + AVG300 + slg_percent +
                    on_base_percent + Position,
  data = stats,
  family = "binomial"
)
tidy(m1)
```

```
# A tibble: 24 x 5
   term                estimate std.error statistic p.value
   <chr>                  <dbl>     <dbl>     <dbl>   <dbl>
 1 (Intercept)         -5.96      2.78       -2.15   0.0319
 2 player_age          -0.0369    0.0570     -0.647  0.518
 3 b_ab                -0.00925   0.00925    -1.00   0.317
 4 b_total_hits         0.0450    0.0300      1.50   0.133
```

```
 5 b_double              0.0182    0.0411      0.443  0.657
 6 b_triple             -0.0469    0.116      -0.404  0.686
 7 b_home_run            0.0719    0.0524      1.37   0.170
 8 b_strikeout          -0.00188   0.00930    -0.203  0.839
 9 b_bb_percent          0.155     0.105       1.48   0.140
10 AVG300Less than 300 -0.617     0.768      -0.804  0.421
# ... with 14 more rows
```

```
m1_aug <- augment(m1) %>%
  mutate(prob = exp(.fitted)/(1 + exp(.fitted)),
         pred_leg = ifelse(prob > 0.32, "All-Star", "Not All-Star"))
table(m1_aug$pred_leg, m1_aug$All.Star)
```

```
              0   1
All-Star     22  30
Not All-Star 410  24
```

```
#Advanced model
m2 <- glm(All.Star ~ player_age + launch_angle_avg +
                     barrel + solidcontact_percent + flareburner_percent +
                     hard_hit_percent + meatball_swing_percent,
  data = stats,
  family = "binomial"
)
tidy(m2)
```

```
# A tibble: 8 x 5
  term                   estimate std.error statistic  p.value
  <chr>                     <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)              1.38       1.76      0.785 4.32e- 1
2 player_age              -0.0361     0.0468   -0.772 4.40e- 1
3 launch_angle_avg        -0.00881    0.0283   -0.312 7.55e- 1
4 barrel                   0.0852     0.0136    6.27  3.56e-10
5 solidcontact_percent    -0.0805     0.0943   -0.854 3.93e- 1
6 flareburner_percent     -0.0129     0.0379   -0.341 7.33e- 1
7 hard_hit_percent        -0.0256     0.0263   -0.974 3.30e- 1
8 meatball_swing_percent -0.0358     0.0162   -2.21  2.72e- 2
```

```
m2_aug <- augment(m2) %>%
  mutate(prob = exp(.fitted)/(1 + exp(.fitted)),
         pred_leg = ifelse(prob > 0.32, "All-Star", "Not All-Star"))
table(m2_aug$pred_leg, m2_aug$All.Star)
```

```
              0   1
All-Star     22  20
Not All-Star 410  34
```

```
# obp percentage lasso
y <- stats$on_base_percent
x <- model.matrix(on_base_percent ~ launch_angle_avg + sweet_spot_percent +
                    barrel + solidcontact_percent + flareburner_percent +
                    hard_hit_percent + avg_hyper_speed + z_swing_percent +
                    oz_swing_percent + meatball_swing_percent, data = stats)
m_lasso_cv <- cv.glmnet(x, y, alpha = 1)
best_lambda <- m_lasso_cv$lambda.min
best_lambda
```

```
[1] 0.0008123019
```

```
m_best <- glmnet(x, y, alpha = 1, lambda = best_lambda)
m_best$beta
```

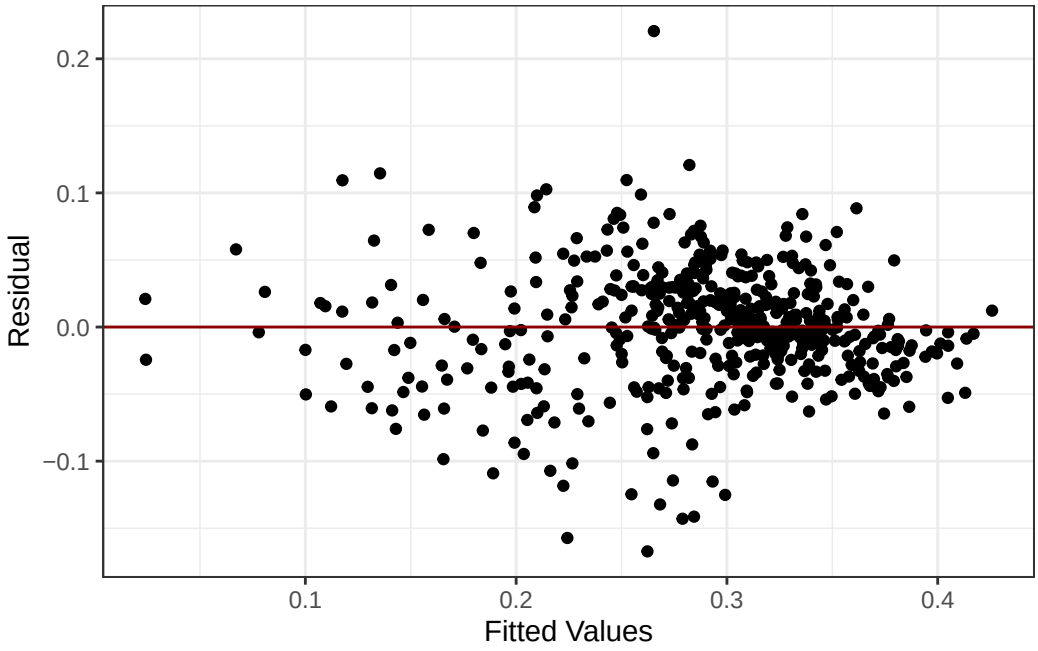
```
11 x 1 sparse Matrix of class "dgCMatrix"
                                    s0
(Intercept)              .
launch_angle_avg        -9.288285e-05
sweet_spot_percent       2.064389e-03
barrel                   1.296901e-03
solidcontact_percent     2.094898e-03
flareburner_percent      3.641493e-03
hard_hit_percent         8.680041e-04
avg_hyper_speed          .
z_swing_percent          5.109151e-04
oz_swing_percent        -2.816476e-03
meatball_swing_percent   7.102024e-04
```

```r
# obp percentage prediction
m3 <- lm(on_base_percent ~ sweet_spot_percent +
                    barrel + solidcontact_percent + flareburner_percent +
                    hard_hit_percent + z_swing_percent +
                    oz_swing_percent + meatball_swing_percent,
  data = stats)

summary(m3)
```

```
Call:
lm(formula = on_base_percent ~ sweet_spot_percent + barrel +
    solidcontact_percent + flareburner_percent + hard_hit_percent +
    z_swing_percent + oz_swing_percent + meatball_swing_percent,
    data = stats)

Residuals:
      Min        1Q    Median        3Q       Max
-0.167277 -0.023263 -0.000016  0.026078  0.220647

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             0.0700061  0.0194563   3.598 0.000354 ***
sweet_spot_percent      0.0019772  0.0004004   4.938 1.09e-06 ***
barrel                  0.0013301  0.0001686   7.891 2.07e-14 ***
solidcontact_percent    0.0022739  0.0008601   2.644 0.008470 **
flareburner_percent     0.0038182  0.0004997   7.641 1.19e-13 ***
hard_hit_percent        0.0008075  0.0002987   2.703 0.007115 **
z_swing_percent         0.0006259  0.0004381   1.428 0.153809
oz_swing_percent       -0.0029453  0.0003171  -9.288  < 2e-16 ***
meatball_swing_percent  0.0007211  0.0002533   2.847 0.004610 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04441 on 477 degrees of freedom
Multiple R-squared:  0.6983,    Adjusted R-squared:  0.6933
F-statistic:   138 on 8 and 477 DF,  p-value: < 2.2e-16
```

```r
m3_aug <- augment(m3)
m3_aug |>
ggplot(aes(x = .fitted, y = .resid)) +
```

```
  geom_point() +
  geom_hline(yintercept = 0, color = "darkred") +
  labs(x = "Fitted Values",
       y = "Residual") +
  theme_bw()
```




```
# slugging percentage lasso
y <- stats$slg_percent
x <- model.matrix(slg_percent ~ launch_angle_avg + sweet_spot_percent +
                    barrel + solidcontact_percent + flareburner_percent +
                    hard_hit_percent + avg_hyper_speed + z_swing_percent +
                    oz_swing_percent + meatball_swing_percent, data = stats)
m_lasso_cv <- cv.glmnet(x, y, alpha = 1)
best_lambda <- m_lasso_cv$lambda.min
best_lambda
```

```
[1] 0.001263641
```

```
m_best <- glmnet(x, y, alpha = 1, lambda = best_lambda)
m_best$beta
```

```
11 x 1 sparse Matrix of class "dgCMatrix"
                                    s0
(Intercept)             .
launch_angle_avg        0.0004816089
sweet_spot_percent      0.0039431166
barrel                  0.0035266068
solidcontact_percent    0.0018502200
flareburner_percent     0.0028240531
hard_hit_percent        0.0014778642
avg_hyper_speed         0.0044117933
z_swing_percent         0.0011833724
oz_swing_percent       -0.0006357843
meatball_swing_percent  0.0006673726
```

```r
# slugging percentage prediction
m4 <- lm(slg_percent ~ launch_angle_avg + sweet_spot_percent +
                    barrel + solidcontact_percent + flareburner_percent +
                    hard_hit_percent + avg_hyper_speed + z_swing_percent +
                    oz_swing_percent + meatball_swing_percent,
  data = stats)

summary(m4)
```

```
Call:
lm(formula = slg_percent ~ launch_angle_avg + sweet_spot_percent +
    barrel + solidcontact_percent + flareburner_percent + hard_hit_percent +
    avg_hyper_speed + z_swing_percent + oz_swing_percent + meatball_swing_percent,
    data = stats)

Residuals:
      Min        1Q    Median        3Q       Max
-0.260804 -0.041484  0.002206  0.040864  0.294753

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)            -0.0807292  0.0316715  -2.549   0.0111 *
launch_angle_avg        0.0005148  0.0005433   0.948   0.3438
sweet_spot_percent      0.0038654  0.0006637   5.824 1.06e-08 ***
barrel                  0.0035645  0.0002677  13.318  < 2e-16 ***
solidcontact_percent    0.0022226  0.0013590   1.635   0.1026
```

```
flareburner_percent     0.0031461  0.0007964   3.950 9.00e-05 ***
hard_hit_percent        0.0011213  0.0010244   1.095   0.2742
avg_hyper_speed         0.0060747  0.0058969   1.030   0.3035
z_swing_percent         0.0013843  0.0006861   2.018   0.0442 *
oz_swing_percent       -0.0008703  0.0005029  -1.731   0.0842 .
meatball_swing_percent  0.0006949  0.0003958   1.756   0.0798 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.06915 on 475 degrees of freedom
Multiple R-squared:  0.7432,    Adjusted R-squared:  0.7378
F-statistic: 137.5 on 10 and 475 DF,  p-value: < 2.2e-16
```

```r
m4_aug <- augment(m4)
m4_aug |>
ggplot(aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "darkred") +
  labs(x = "Fitted Values",
       y = "Residual") +
  theme_bw()
```