

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.

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. The final data file we have was compiled from baseballsavant.com with a mix of more traditional stats and statcast stats. This complete file can be found in the stats.csv file. Once that was done, we entered a player's position, salary, and team from baseball prospectus. we used wikipedia to find rosters for the 2019 all-star game and created a categorical variable column with that information. ## 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", pi
    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] 0.0002836883
```

```
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                    -0.002884964
b_ab                          -0.001912416
b_total_pa                    .
b_total_hits                  0.005156937
b_home_run                    0.009793950
AVG300Less than 300          -0.127962775
batting_avg                    .
b_double                      0.002591757
b_triple                      0.004050357
HR40Less than 40             -0.180334732
b_strikeout                  -0.000599696
b_walk                        0.003621178
slg_percent                   0.073240458
on_base_percent              -0.417590129
Position2B                    0.047178572
Position3B                   -0.030771133
PositionC                     0.071887713
PositionCF                    0.060519117
PositionCH                    0.029243371
PositionDH                   -0.037514621
PositionDNP                  -0.005012120
PositionLF                   -0.048210565
PositionPH                    0.009621017
PositionRF                    0.006668194
PositionSP                    0.166475592
PositionSS                    0.051044440
AVG300Less than 300:batting_avg 0.044192051
b_home_run:HR40Less than 40   0.002048005

```

```

# 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] 0.005820234
```

```
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      -0.0018096273
launch_angle_avg -0.0001599061
sweet_spot_percent .
barrel          0.0080466625
solidcontact_percent -0.0030825651
flareburner_percent -0.0018259472
hard_hit_percent  -0.0018130937
avg_hyper_speed   .
z_swing_percent   .
oz_swing_percent  .
meatball_swing_percent -0.0021930005
```

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] 0.001293414
```

```

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  .
sweet_spot_percent 0.0020557555
barrel            0.0012787619
solidcontact_percent 0.0019445642
flareburner_percent 0.0035450332
hard_hit_percent  0.0009006682
avg_hyper_speed    .
z_swing_percent    0.0004246729
oz_swing_percent   -0.0027211972
meatball_swing_percent 0.0006951876

```

```
# 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

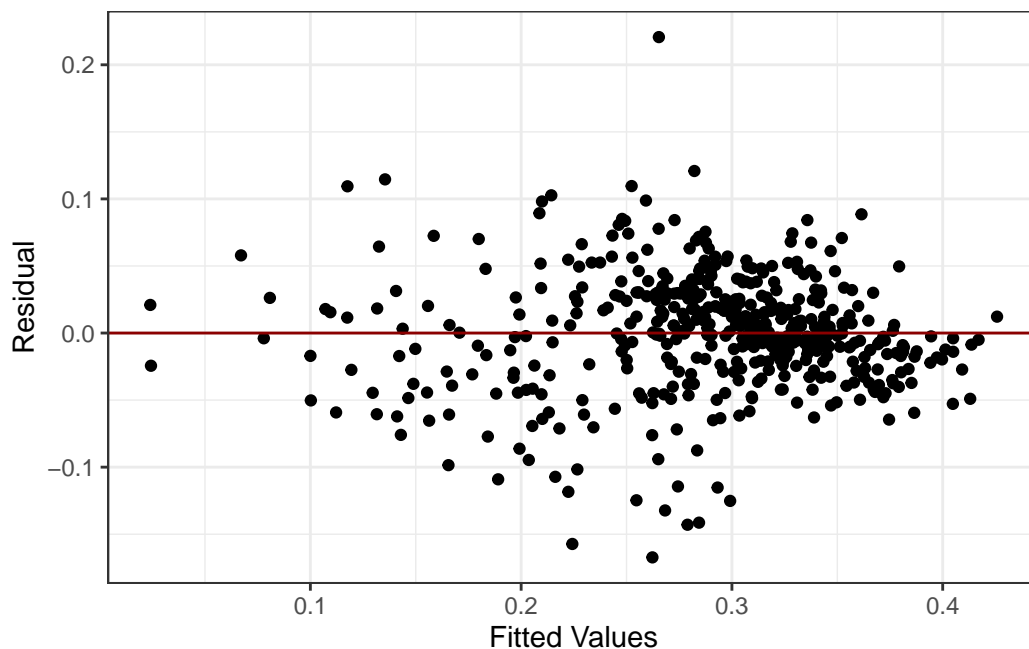
Residual standard error: 0.04441 on 477 degrees of freedom

Multiple R-squared: 0.6983, Adjusted R-squared: 0.6933

F-statistic: 138 on 8 and 477 DF, p-value: < 2.2e-16

```
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] 0.004235222
```

```
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  4.043655e-04
sweet_spot_percent 4.127838e-03
barrel            3.439947e-03
solidcontact_percent 1.017289e-03
flareburner_percent 2.091817e-03
hard_hit_percent  2.230822e-03
avg_hyper_speed    9.253061e-04
z_swing_percent    7.147313e-04
oz_swing_percent   -9.204756e-05
meatball_swing_percent 6.042334e-04
```

```
# 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 .

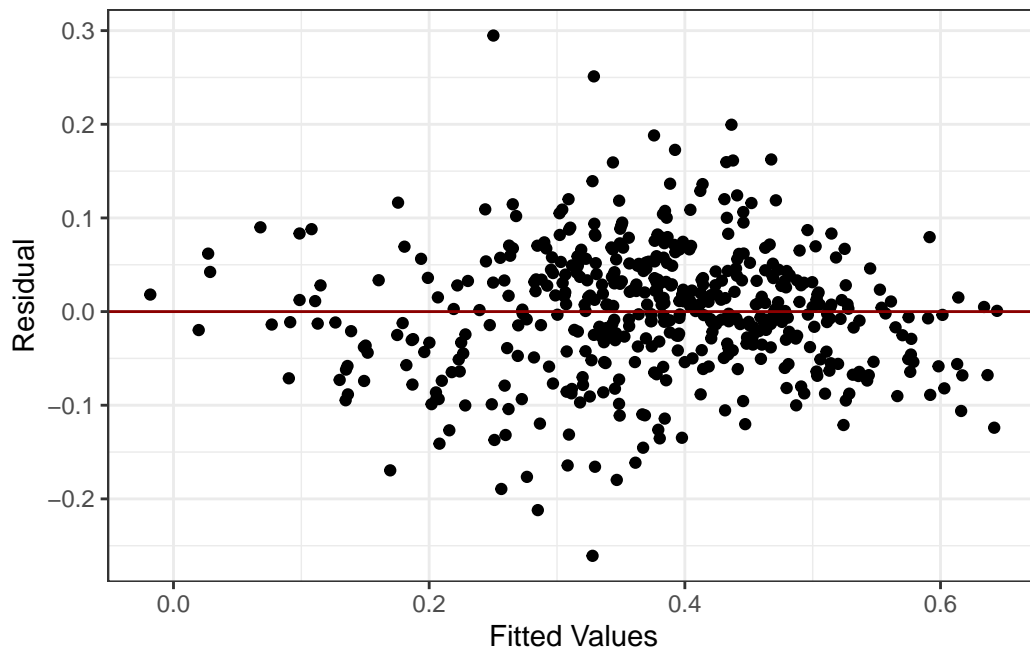
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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

```
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()
```



```

# Subset for nationals
nationals_stats <- stats |>
  filter(Team == "WAS")

# Predict
pred_obp <- predict(m3, nationals_stats)
pred_slg <- predict(m4, nationals_stats)

# Add to DF
nationals_stats <- nationals_stats |>
  mutate(Predicted_OBP = pred_obp,
         Predicted_SLG = pred_slg)

# Display
print(nationals_stats)

```

	player_id	year	player_age	b_ab	b_total_pa	b_total_hits	b_double	b_triple
1	434671	2019	35	52	56	6	0	0
2	435062	2019	35	334	370	115	23	1
3	435559	2019	35	280	309	74	11	0
4	453286	2019	34	55	61	10	0	0
5	475582	2019	34	171	190	44	9	0
6	476451	2019	32	9	13	1	0	0
7	543228	2019	31	314	358	70	16	0
8	543685	2019	29	545	646	174	44	3
9	544931	2019	30	72	80	12	1	0
10	571431	2019	30	310	333	70	14	0
11	571578	2019	29	65	72	6	1	0
12	572191	2019	28	88	97	22	7	0
13	572821	2019	32	416	482	99	20	0
14	594694	2019	27	131	144	33	2	0
15	594809	2019	30	566	656	158	25	7
16	607208	2019	26	521	569	155	37	5
17	645302	2019	22	546	617	139	33	3
18	664057	2019	25	30	37	11	1	1
19	665742	2019	20	542	659	153	32	5
20	669738	2019	25	12	13	2	1	0

	b_home_run	b_strikeout	b_walk	b_k_percent	b_bb_percent	batting_avg
1	0	24	0	42.9	0.0	0.115
2	17	49	27	13.2	7.3	0.344
3	17	36	20	11.7	6.5	0.264
4	0	27	0	44.3	0.0	0.182

5	6	39	17	20.5	8.9	0.257			
6	0	4	2	30.8	15.4	0.111			
7	12	84	38	23.5	10.6	0.223			
8	34	86	80	13.3	12.4	0.319			
9	1	25	3	31.3	3.8	0.167			
10	20	115	20	34.5	6.0	0.226			
11	0	27	3	37.5	4.2	0.092			
12	1	34	7	35.1	7.2	0.250			
13	20	105	61	21.8	12.7	0.238			
14	2	29	12	20.1	8.3	0.252			
15	15	106	65	16.2	9.9	0.279			
16	19	113	43	19.9	7.6	0.298			
17	17	140	35	22.7	5.7	0.255			
18	0	11	6	29.7	16.2	0.367			
19	34	132	108	20.0	16.4	0.282			
20	0	4	1	30.8	7.7	0.167			
	slg_percent	on_base_percent	xba	xslg	woba	xwoba	xobp	xiso	xslgdiff
1	0.115	0.107	0.127	0.146	0.100	0.121	0.127	0.019	-0.031
2	0.572	0.395	0.332	0.614	0.400	0.418	0.388	0.282	-0.042
3	0.486	0.324	0.277	0.455	0.337	0.338	0.337	0.178	0.031
4	0.182	0.164	0.152	0.188	0.158	0.147	0.152	0.035	-0.006
5	0.415	0.321	0.248	0.444	0.313	0.325	0.316	0.196	-0.029
6	0.111	0.231	0.099	0.110	0.205	0.200	0.263	0.011	0.001
7	0.389	0.316	0.217	0.391	0.298	0.302	0.313	0.174	-0.002
8	0.598	0.412	0.310	0.590	0.413	0.414	0.407	0.280	0.008
9	0.222	0.188	0.191	0.279	0.186	0.221	0.224	0.088	-0.057
10	0.465	0.276	0.228	0.479	0.306	0.320	0.281	0.251	-0.014
11	0.108	0.125	0.106	0.133	0.112	0.131	0.146	0.028	-0.025
12	0.364	0.299	0.213	0.375	0.289	0.276	0.271	0.163	-0.011
13	0.430	0.340	0.235	0.426	0.329	0.330	0.339	0.191	0.004
14	0.313	0.313	0.246	0.319	0.270	0.272	0.310	0.073	-0.006
15	0.428	0.360	0.256	0.406	0.342	0.328	0.346	0.149	0.022
16	0.497	0.353	0.279	0.455	0.356	0.339	0.337	0.176	0.042
17	0.419	0.323	0.231	0.368	0.317	0.294	0.305	0.137	0.051
18	0.467	0.486	0.216	0.388	0.417	0.339	0.369	0.172	0.079
19	0.548	0.401	0.285	0.575	0.394	0.409	0.405	0.290	-0.027
20	0.250	0.231	0.176	0.199	0.214	0.202	0.239	0.023	0.051
	exit_velocity_avg	launch_angle_avg	sweet_spot_percent	barrel					
1	78.6	-1.5		25.0	0				
2	91.7	10.8		36.9	33				
3	86.0	18.9		39.7	16				
4	87.2	-7.5		20.6	0				
5	91.4	9.7		32.8	10				

6	76.7	-12.0	0.0	0
7	87.6	18.1	32.0	16
8	90.4	19.5	38.7	56
9	88.0	-3.9	26.9	1
10	88.6	19.6	39.8	29
11	81.9	-5.8	16.7	0
12	91.5	8.9	32.1	4
13	88.5	19.0	32.4	21
14	83.9	10.1	34.0	0
15	88.9	15.3	33.9	18
16	90.4	9.9	29.8	28
17	83.3	16.7	34.5	20
18	84.6	12.0	31.6	2
19	92.0	12.5	36.3	51
20	78.0	-2.8	25.0	0
solidcontact_percent flareburner_percent hard_hit_percent avg_hyper_speed				
1	0.0	15.6	0.0	1.041144
2	8.6	27.2	48.3	7.726170
3	4.5	29.1	32.4	4.618345
4	0.0	23.5	20.6	4.216138
5	11.2	19.4	49.3	7.398411
6	0.0	14.3	14.3	3.160608
7	6.5	22.5	36.1	5.052168
8	6.6	27.4	46.6	6.364730
9	1.9	26.9	28.8	4.335569
10	8.2	18.9	43.9	6.825084
11	2.4	16.7	11.9	1.749437
12	8.9	23.2	42.9	7.216883
13	7.1	26.9	35.4	5.121792
14	5.8	27.2	19.4	2.731144
15	6.6	26.3	34.3	5.472874
16	6.3	26.1	42.0	6.839620
17	5.5	21.3	23.0	3.739896
18	5.3	21.1	31.6	4.248422
19	8.7	25.5	47.8	7.569585
20	0.0	25.0	0.0	0.500720
z_swing_percent oz_swing_percent meatball_swing_percent Salary Position				
1	50.9	56.4	50.0	8000000 SP
2	64.0	31.4	72.2	4000000 1B
3	74.2	29.7	83.1	4000000 C
4	66.4	34.7	68.0	37405562 SP
5	56.7	27.0	52.1	18000000 1B
6	44.1	13.0	80.0	1300000 SP

7	67.2	34.3	76.3	7083333	C
8	68.3	20.6	78.6	18800000	3B
9	62.4	29.6	52.4	38333334	SP
10	70.5	34.1	76.5	3000000	1B
11	58.6	41.5	63.6	12916666	SP
12	69.7	24.7	75.0	3250000	CF
13	62.2	19.6	69.8	9000000	2B
14	79.0	30.4	92.2	581100	SS
15	63.8	27.4	71.4	8400000	RF
16	67.4	29.2	70.5	3725000	SS
17	71.1	30.1	82.6	557800	CF
18	76.3	31.7	80.0	559100	PH
19	66.7	20.3	79.9	578300	LF
20	68.4	50.0	66.7	555000	PH

	Team	All.Star	last_name	first_name		AVG300	HR40	pitcher
1	WAS	0	Sanchez	Anibal	Less than	300	Less than 40	Yes
2	WAS	0	Kendrick III	Howie	Greater than	300	Less than 40	No
3	WAS	0	Suzuki	Kurt	Less than	300	Less than 40	No
4	WAS	1	Scherzer	Max	Less than	300	Less than 40	Yes
5	WAS	0	Zimmerman	Ryan	Less than	300	Less than 40	No
6	WAS	0	Hellickson	Jeremy	Less than	300	Less than 40	Yes
7	WAS	0	Gomes	Yan	Less than	300	Less than 40	No
8	WAS	1	Rendon	Anthony	Greater than	300	Less than 40	No
9	WAS	0	Strasburg	Stephen	Less than	300	Less than 40	Yes
10	WAS	0	Adams	Matt	Less than	300	Less than 40	No
11	WAS	0	Corbin	Patrick	Less than	300	Less than 40	Yes
12	WAS	0	Taylor	Michael A.	Less than	300	Less than 40	No
13	WAS	0	Dozier	Brian	Less than	300	Less than 40	No
14	WAS	0	Difo	Wilmer	Less than	300	Less than 40	No
15	WAS	0	Eaton	Adam	Less than	300	Less than 40	No
16	WAS	0	Turner	Trea	Less than	300	Less than 40	No
17	WAS	0	Robles	Victor	Less than	300	Less than 40	No
18	WAS	0	Stevenson	Andrew	Greater than	300	Less than 40	No
19	WAS	0	Soto	Juan	Less than	300	Less than 40	No
20	WAS	0	Noll	Jake	Less than	300	Less than 40	No

	Predicted_OBP	Predicted_SLG
1	0.08079562	0.12665655
2	0.34890606	0.50231016
3	0.33617350	0.44003788
4	0.20548758	0.22665050
5	0.28104218	0.36403902
6	0.18315099	0.09864245
7	0.28045157	0.38190917

8	0.41701544	0.60153597
9	0.24446703	0.26932063
10	0.31238228	0.47601924
11	0.14216221	0.15187871
12	0.30721168	0.37400951
13	0.34097188	0.41791040
14	0.29633013	0.33967319
15	0.31481301	0.40554899
16	0.32108186	0.44099774
17	0.29263481	0.38988303
18	0.26535264	0.32775106
19	0.40492893	0.57713739
20	0.15853137	0.19367725

ARTICLE ABOUT BATTING ORDER STRATEGY: <https://www.sportsbettingdime.com/guides/strategy/batting-order-sabermetrics/>