SkwOBA and OOE

A Skill-Based Decomposition of Pitcher xwOBA

Ben Gratz | Washington University in St. Louis | B.A. Economics & Computer Science, 2027

2015–2025 MLB Seasons, with a focus on the 107 qualified starting pitchers from the 2025 season | Data: Baseball Savant | Coding and calculations performed in STATA

SkwOBA and OOE: A Skill-Based Decomposition of Pitcher xwOBA

This project was completed using data from Baseball Savant on pitchers from 2015 to 2025, with a focus on the 107 qualified starting pitchers from the 2025 MLB season. Coding and calculations were performed using STATA.

As the prevalence of sabermetrics and advanced statistics in baseball rises, I wanted to more closely examine factors that drive pitcher success in the MLB. Therefore, I created a statistical decomposition of xwOBA with a focus on repeatable, skill-based variables. This will allow us to better understand what traits or skill sets make pitchers successful. Furthermore, I will investigate trends regarding the difference between the xwOBA decomposition and true xwOBA to analyze over/under performance based on expected outcomes.

To begin, it is important to have a strong understanding of the xwOBA metric. xwOBA (expected weighted on-base average) estimates expected performance of both hitters and pitchers based on Statcast metrics such as exit velocity and launch angle. The Statcast metrics of balls in play are used to calculate probabilities of given results (e.g. singles, home runs, etc.) and these probabilities are multiplied by given constants. xwOBA differs from wOBA in the fact that wOBA only uses real outcomes, rather than probabilities based off of Statcast metrics. While it is statistically powerful that xwOBA captures expected outcomes, a model that is focused on outcomes rather than inputs can make it difficult to predict success. So, my goal was to create a simplified decomposition of xwOBA that captures much of the sentiment of xwOBA, while tracking underlying variables in the pitcher's control.

Strikeouts and walks are the two results in baseball that the pitcher is the most in control of. Aside from rare missed calls (which will likely become fewer with the implementation of the ABS system), the pitcher is in complete control. Balls in play become more complicated because they involve defense, shifts, ballpark factors, and more. So, I knew that it was crucial to include strikeouts and walks in this metric. To test my intuition of the importance of strikeouts and walks, I performed an auxiliary linear regression of a stat that I labeled "kbb" (K% - BB%) on xwOBA. I elected to use subtraction rather than division as is seen in a typical K:BB ratio in order to limit outliers with particularly high or low walk rates, such as Zach Littell at 4.2% or Dylan Cease at 9.8%. As suspected, the R-squared value of this regression was rather high, at .7236, meaning that subtracting walk percentage from strikeout percentage explains over 72% in xwOBA variance by itself. Results can be found in Image E1 in the appendix.

After studying kbb, I knew that there were still more outcomes that I had to factor into my evaluation. Thus, I shifted my focus towards contact. While many metrics, including xwOBA, are focused on the results of contact, such as types of hits, I wanted a model that was largely fielding and luck independent. I decided that there were two main aspects of contact that are most important to pitching success – missing barrels and inducing weak contact. Missing barrels is essential for limiting damage, and inducing weak contact may be among the most important skills for pitchers. As the game and baseball analytics shift to favor strikeout rates and whiff percentages, I believe there is still value to pitchers who induce weak contact. Drawing weak contact, especially as a starting pitcher, is important for keeping pitch count low and getting deep into games. Thus, I regressed z_poorlytopped, z_poorlyunder, z_kbb, and z_barrel against z_xwoba.

The z's represent that each variable is a standardization, and each of the regressors were measured as percentages or rates. This allows for easy computations across different metrics and scales. "Poorly under" contact represents softly hit pop ups where a batter gets under the ball, while "poorly topped" contacts are typically weak ground balls. Batted barrel rate takes barrels (calculated using exit velocity and launch angle thresholds associated with high run values) divided by balls put in play. A complete list of variable and metric definitions can be found in the appendix in Table A1.

Although the purpose of this decomposition is to move away from outcome-based results such as these, tracking barrels and weak contacts is critical in representing repeatable skill sets. In xwOBA, a ball that is hit weakly could still count against a pitcher if historically similar contacts have often resulted in hits (e.g. a poorly topped groundball between the shortstop and third baseman). In addition to weak contact, barrel rates play an important role in this metric because they account for the other end of the contact spectrum. A contact-inducing pitcher can appear successful with impressive weak contact metrics, but this entirely ignores the hard contact that they could potentially allow. Therefore, barrel rate's inclusion in this model affords us a deeper understanding of a pitcher's contact profile.

This regression (Image E2) had an R-squared value of .936, meaning that weak contact (composed of poorly topped and poorly under contacts), kbb, and barrel rate nearly perfectly captures the variance of xwOBA, while focusing on largely repeatable, skill-based measures. Furthermore, each component has a p-value of 0.000, demonstrating a statistical significance beyond the 1% level. I used the coefficients from this regression to turn these variables into a calculatable statistic. Introducing Skill-Based Weighted On-Base Average (SkwOBA, pronounced "skwoh-bah"):

Using the coefficients from the data on the 2025 season, I calculated:

$$SkwOBA = ((.5962999*z\_poorlytopped) + (.6124874*z\_poorlyunder) + (.7026914*z\_kbb) + (-.2830242*z\_barrel))$$

In addition, I multiplied the equation by -1, so that a higher SkwOBA is better whereas a higher pitcher xwOBA is worse. It is important to note that due to different statistics across different seasons, these coefficients will likely change. A new regression should be run each season or time frame being studied to exactly find the correct coefficients, or using the above coefficients as an estimate would likely suffice. The top and bottom 10 SkwOBAs of 2025 pitchers are listed below. The full table can be found in the appendix (Table B1).

Rank	**Name**	**SkwOBA**
1	Wheeler, Zack	2.563083
2	Sale, Chris	2.379977
3	Skenes, Paul	2.294365
4	Skubal, Tarik	2.045081
5	Gilbert, Logan	2.015501
6	Yamamoto, Yoshinobu	1.865069
7	Sánchez, Cristopher	1.593681
8	Crochet, Garrett	1.496935
9	Brown, Hunter	1.486108
10	Luzardo, Jesús	1.371883
...	...	...
98	Buehler, Walker	-1.36522
99	Vásquez, Randy	-1.44169
100	Heaney, Andrew	-1.45801
101	Irvin, Jake	-1.50588
102	Fedde, Erick	-1.5181
103	Kochanowicz, Jack	-1.64149
104	Márquez, Germán	-1.6785
105	Quantrill, Cal	-1.79344
106	Parker, Mitchell	-1.82555
107	Senzatela, Antonio	-1.99693

Once I mathematically defined SkwOBA, I was eager to explore its applications. I believe the greatest strength of SkwOBA to be its reduction in the influence of luck and end results, instead focusing on a pitcher's skill profile. Although similar to xwOBA, these two metrics serve different purposes. xwOBA measures expected run prevention based on outcomes, while SkwOBA analyzes the underlying pitcher traits that lead to the results tracked by xWOBA.

Predicting future outcomes is one of the most important yet most difficult tasks in sabermetrics. Given that SkwOBA is a decomposition of xwOBA based on key skills, I was curious to test how SkwOBA predicts future results compared to the outcome-driven xwOBA. Therefore, I calculated SkwOBA for each year since the introduction of Statcast in 2015 (excluding the COVID-shortened 2020 season) by regressing the same five variables against xwOBA. These yearly SkwOBA formulas can be found in the Section D of the appendix. As suspected, the coefficients showed slight variations between years. However, it was interesting to find that the R^2 value for each of these regressions was between 0.88 and 0.94, demonstrating that the results from 2025 are not an anomaly and that these variables do capture much of the sentiment of xwOBA. I then regressed SkwOBA against the following year's xwOBA and compared these results to a regression of xwOBA against itself year over year. The results were fascinating. Across 8 seasons, SkwOBA proved to be a better predictor of xwOBA than lagged xwOBA in 5 out of 8 seasons. I have attached summaries of my comparisons of the predictive power of SkwOBA and xwOBA lag using the R^2 and root mean square errors (RMSE) to measure results. The season listed is the season's xwOBA which the model is attempting to predict using the previous year's statistics. Complete regression results can be found in the appendix in images E6-E20.

Season	**SkwOBA R^2**	**xwOBA Lag R^2**	**SkwOBA RMSE**	**xwOBA Lag RMSE**	**Better R^2 Model**	**Better RMSE Model**
2016	0.3588	0.2938	0.02161	0.02268	SkwOBA	SkwOBA
2017	0.3284	0.3421	0.02426	0.02401	xwOBA	xwOBA
2018	0.2781	0.2331	0.02553	0.02631	SkwOBA	SkwOBA
2019	0.3257	0.3226	0.02313	0.02318	SkwOBA	SkwOBA
2022	0.2788	0.2758	0.02539	0.02544	SkwOBA	SkwOBA
2023	0.2204	0.2432	0.02175	0.02143	xwOBA	xwOBA
2024	0.3486	0.3613	0.02052	0.02032	xwOBA	xwOBA
2025	0.4511	0.4414	0.02076	0.02094	SkwOBA	SkwOBA

Metric	**SkwOBA**	**xwOBA Lag**
Average R^2	0.3237	0.3142
Average RMSE	0.0229	0.023

As we see, the R^2 values of both SkwOBA and xwOBA hover between 0.2 to 0.45, indicating just how difficult it is to predict future success. Although the edge is small and we're working with a limited sample size, it appears that SkwOBA could be a marginally better (or at least as good) predictor of future success than xwOBA. Some may argue that SkwOBA producing similar predictive results to lagged xwOBA is not impressive, given that SkwOBA is essentially a decomposition of xwOBA. To an extent this is true; I would at least expect results to be somewhat similar. However, as discussed above, SkwOBA captures about 90% of xwOBA variance. So, although quite similar, they are not identical. The fact that SkwOBA still demonstrates a slight predictive edge suggests that SkwOBA has potentially eliminated some variation that is captured by xwOBA which is commonly associated with probabilistic outcome modeling. This indicates that emphasizing underlying skill traits, such as strikeout to walk ratios and contact profiles, may represent a more stable evaluation of pitcher skill. Finally, I attempted to regress both SkwOBA and xwOBA on future xwOBA but found that using the two in combination adds no significant predictive value and that neither variable is statistically significant (Image E21). This result is expected given that there is such a high overlap in the information conveyed between the two metrics.

Next, I wanted to compare this xwOBA-inspired stat to pitchers' actual xwOBA. As previously mentioned, I negated xwOBA so that, like SkwOBA, a higher value indicates better pitcher performance. I subtracted the SkwOBA z-score from the xwOBA z-score to yield what I called "Outcome Over Expected", or OOE (pronounced "oh-oh-ee"). The idea of this OOE metric is simple. Pitchers who have higher xwOBAs than SkwOBAs will have positive OOE scores. I interpret pitchers who score highly in OOE to have overperformed given their underlying skill. While this is a simplification, Outcome Over Expected can be interpreted as a proxy for "luck", where higher values reflect more favorable outcomes relative to underlying skill. If a pitcher has an xwOBA better than their SkwOBA, they benefited from favorable outcomes such as exceptional defense or hard-hit contacts resulting in outs. Furthermore, a pitcher with a positive OOE could be benefitting from sequencing and strand rate luck, meaning they were fortunate in that much of their damage occurs with the bases empty. On the other hand, a pitcher with a negative OOE means that their xwOBA shows that they are worse than their SkwOBA indicates. These are the cases where pitchers get unlucky with bad defensive range (not errors) or weak contacts finding holes. According to SkwOBA, these pitchers are better than their xwOBA would indicate. I have copied the table of select pitchers in 2025 and their OOE, sorted from highest to lowest, where I believe the pitchers with a higher OOE had a worse season than their xwOBA would indicate, while pitchers with low OOE performed better than many numbers show. Select OOE scores are shown for the 2025 season. The complete table can be found in the appendix (Table C1).

Rank	**Name**	**OOE**
1	Allen, Logan	0.6851459
2	Priester, Quinn	0.6810052
3	Kremer, Dean	0.4866926
4	Boyd, Matthew	0.3765679
5	Ray, Robbie	0.3686397
6	Pepiot, Ryan	0.3549854
7	Soriano, José	0.3538436
8	Fedde, Erick	0.3369309
9	Rea, Colin	0.3268954
10	Gausman, Kevin	0.3264024
...	...	...
58	Skubal, Tarik	0.0002885
...	...	...
98	Smith, Shane	-0.3719033
99	Cameron, Noah	-0.3722444
100	Pfaadt, Brandon	-0.3753321
101	May, Dustin	-0.39689
102	Sugano, Tomoyuki	-0.4607614
103	Strider, Spencer	-0.4991508
104	Lorenzen, Michael	-0.522287
105	Gilbert, Logan	-0.5327203
106	Sale, Chris	-0.5572051
107	Senzatela, Antonio	-0.6827681

Note: OOE scores are measured in standard deviations

There were some particular instances in this data that I found interesting and wanted to highlight. First, I wanted to investigate the names at the top of this list, pitchers who seemingly got lucky or overperformed their underlying skill. Logan Allen and Quinn Priester are the top two respectively. In 2025, both took huge jumps from the previous season. Allen improved his FIP from 5.87 to 4.37, while Priester went from 4.70 to 4.01. However, these extremely high OOEs tell us not to jump to any conclusions, as some of this improvement could be attributed to luck. Also near the top of this list, I was surprised to find Robbie Ray (5th) and Kevin Gausman (10th), who have been among the best starters in baseball for the last 5+ years. Next, moving to the middle of the list, I was intrigued to find that Tarik Skubal was the player with an OOE closest to 0, meaning that his xwOBA and his SkwOBA are very aligned. Skubal's dominance cannot be attributed to luck, as his underlying skill and results line up nearly perfectly. Interestingly, there were some all-star caliber pitchers at the bottom of this list. For example, Logan Gilbert and Chris Sale (numbers 105 and 106 respectively) both had great seasons, but these numbers indicate that they performed even better than one might think. Finally, the bottom name, Antonio Senzatela, had the highest xwOBA among qualified pitchers at a whopping 0.395. This put him nearly 3 standard deviations below the mean, making his SkwOBA that was only 2 standard deviations below the mean seem better than it is. The story that OOE tells is not necessarily that Senzatela got unlucky all year, simply that he should have had a slightly better season than some other numbers would indicate.

Next, I wanted to investigate if OOE, like SkwOBA, could be used to predict future outcomes. Does a negative OOE, proxying for bad luck in the past, predict future success? If this were the case, these pitchers might make good "buy low" targets for front offices, whereas teams should be wary of pitchers with positive OOEs. Therefore, I used similar framework to calculate SkwOBA and OOE for 2024 and regressed lagged OOE against xwOBA. The resulting R-squared value of 0.001 indicates that OOE has nearly no predictive power on future success (Image E4). Next, I was curious to see if OOE could at least predict a change in xwOBA, season over season. I regressed lagged OOE against xwOBA_diff, where xwOBA_diff is defined as xwoba2025 - xwoba2024 (Image E5). The R-squared value of 0.028 confirms that OOE has very little predictive relevance. However, this result is intuitive. OOE is a luck-based measure of over or underperformance relative to underlying skill level. Therefore, we should not expect that in the following year, a player's fortunes are bound to change from the previous season. However, that does not mean that Outcome Over Expected is useless. While it can't be used to forecast future results, it is still a strong indicator of over or underperformance based on factors we have simplified as "luck". Studying this metric allows us to assess which pitchers had better or worse seasons than their xwOBA might indicate.

In conclusion, SkwOBA is a skill-based decomposition of xwOBA that evaluates pitchers based on what they can control: strikeouts, walks, and contact types. It measures the traits that contribute to xwOBA, while not directly focusing on expected outcomes. While capturing around 90% of the variance of xwOBA, SkwOBA has maintained potentially better predictive power than xwOBA itself, indicating successful elimination of noise and randomness that appears in outcome-driven metrics such as xwOBA. Finally, the calculation of OOE by contrasting SkwOBA with xwOBA is useful to find pitchers who have had more or less success than their skill profile would expect, but due to the randomness of luck, which it proxies for, it holds no real predictive power.

Appendix

A. Variable Definitions

Table A1

Variable	Definition	Source
xwOBA	Expected weighted on-base average based on Statcast data. Assigns run values to batted balls using exit velocity and launch angle to estimate probabilities of outcomes (singles, doubles, HR, etc.), rather than using actual results.	Baseball Savant
k_percent (K%)	Percentage of plate appearances that result in a strikeout (strikeouts ÷ batters faced).	Baseball Savant
bb_percent (BB%)	Percentage of plate appearances that result in a walk (walks ÷ batters faced).	Baseball Savant
kbb	Strikeout rate minus walk rate (K% − BB%), used as a linear measure of pitcher dominance and control.	Derived

Variable	Definition	Source
poorlytopped_percent	Percentage of batted balls classified as "poorly topped," meaning weak ground balls typically hit at low exit velocities and very low launch angles (often resulting in slow rollers or easy groundouts).	Statcast
poorlyunder_percent	Percentage of batted balls classified as "poorly under," meaning weak pop-ups or high fly balls hit with low exit velocity and excessive launch angle (generally easy outs for fielders).	Statcast
barrel_batted_rate	Percentage of batted balls that are "barrels." A barrel is defined by Statcast as a batted ball with a combination of exit velocity and launch angle that historically produces a minimum .500 batting average and 1.500 slugging percentage.	Statcast
z_ variables	Standardized versions of variables, calculated by subtracting the mean and dividing by the standard deviation within a given season.	Derived
SkwOBA	Skill-based weighted on-base average calculated as a weighted combination of standardized K-BB%, weak contact rates, and barrel rate using regression-derived coefficients. Higher values indicate better pitcher performance.	Derived
OOE	Outcome Over Expected, defined as z_xwOBA minus z_SkwOBA. Represents the difference between expected outcomes and underlying skill, often interpreted as a proxy for favorable or unfavorable results.	Derived

B. 2025 SkwOBAs

Table B1

Rank	Name	SkwOBA
1	Wheeler, Zack	2.563083
2	Sale, Chris	2.379977
3	Skenes, Paul	2.294365
4	Skubal, Tarik	2.045081
5	Gilbert, Logan	2.015501
6	Yamamoto, Yoshinobu	1.865069
7	Sánchez, Cristopher	1.593681
8	Crochet, Garrett	1.496935
9	Brown, Hunter	1.486108
10	Luzardo, Jesús	1.371883
11	deGrom, Jacob	1.304572
12	Woo, Bryan	1.261413
13	Suárez, Ranger	1.245867
14	Peralta, Freddy	1.109308
15	Ryan, Joe	1.105602

Rank	Name	SkwOBA
16	Lodolo, Nick	1.015786
17	Webb, Logan	0.93474
18	Rasmussen, Drew	0.929708
19	Fried, Max	0.928943
20	Rodón, Carlos	0.848451
21	Cease, Dylan	0.731936
22	Valdez, Framber	0.641495
23	Kirby, George	0.616066
24	Smith, Shane	0.599
25	Bibee, Tanner	0.59668
26	Taillon, Jameson	0.579027
27	Cameron, Noah	0.531172
28	Abbott, Andrew	0.509162
29	Nelson, Ryne	0.483195
30	Gray, Sonny	0.470259
31	Mize, Casey	0.455686
32	Pivetta, Nick	0.442692
33	Houser, Adrian	0.431668
34	Patrick, Chad	0.396354
35	Gausman, Kevin	0.298264
36	Ray, Robbie	0.291478
37	Baz, Shane	0.262813
38	Sears, JP	0.259502
39	Boyd, Matthew	0.21565
40	Kelly, Merrill	0.181062
41	Cabrera, Edward	0.168616
42	Martinez, Nick	0.130355
43	Priester, Quinn	0.125583
44	Bassitt, Chris	0.121647
45	Hendricks, Kyle	0.11827
46	Singer, Brady	0.100055
47	Bradley, Taj	0.097229
48	Kremer, Dean	0.040949
49	Keller, Mitch	0.007431

Rank	Name	SkwOBA
50	Pepiot, Ryan	-0.00202
51	Lorenzen, Michael	-0.00524
52	Wacha, Michael	-0.00542
53	Imanaga, Shota	-0.00756
54	Williams, Gavin	-0.08843
55	Ober, Bailey	-0.1028
56	Flaherty, Jack	-0.10491
57	Springs, Jeffrey	-0.13378
58	Castillo, Luis	-0.13542
59	Bello, Brayan	-0.13699
60	Littell, Zack	-0.18459
61	Holmes, Clay	-0.18687
62	Severino, Luis	-0.19353
63	Rodriguez, Eduardo	-0.19783
64	Soriano, José	-0.20539
65	Gallen, Zac	-0.24139
66	Alcantara, Sandy	-0.24617
67	Strider, Spencer	-0.26617
68	Gore, MacKenzie	-0.27305
69	Kikuchi, Yusei	-0.28111
70	Morton, Charlie	-0.29055
71	Verlander, Justin	-0.29783
72	Walker, Taijuan	-0.31829
73	Pallante, Andre	-0.32425
74	Elder, Bryce	-0.32927
75	Corbin, Patrick	-0.40756
76	Peterson, David	-0.41643
77	Leiter, Jack	-0.43751
78	Berríos, José	-0.45185
79	Lord, Brad	-0.64301
80	Liberatore, Matthew	-0.64769
81	May, Dustin	-0.67182
82	Warren, Will	-0.68943
83	Rea, Colin	-0.72458

Rank	Name	SkwOBA
84	Giolito, Lucas	-0.75066
85	Falter, Bailey	-0.81777
86	Freeland, Kyle	-0.82557
87	Pfaadt, Brandon	-0.86307
88	Martin, Davis	-0.86751
89	Burke, Sean	-0.9012
90	Allen, Logan	-0.93486
91	Paddack, Chris	-0.93716
92	Quintana, Jose	-0.98873
93	Cecconi, Slade	-1.07303
94	Mikolas, Miles	-1.09864
95	Anderson, Tyler	-1.24883
96	Sugano, Tomoyuki	-1.25752
97	Lugo, Seth	-1.2613
98	Buehler, Walker	-1.36522
99	Vásquez, Randy	-1.44169
100	Heaney, Andrew	-1.45801
101	Irvin, Jake	-1.50588
102	Fedde, Erick	-1.5181
103	Kochanowicz, Jack	-1.64149
104	Márquez, Germán	-1.6785
105	Quantrill, Cal	-1.79344
106	Parker, Mitchell	-1.82555
107	Senzatela, Antonio	-1.99693

C. 2025 OOEs

Table C1

Rank	Name	OOE
1	Allen, Logan	0.6851459
2	Priester, Quinn	0.6810052
3	Kremer, Dean	0.4866926
4	Boyd, Matthew	0.3765679
5	Ray, Robbie	0.3686397
6	Pepiot, Ryan	0.3549854

Rank	Name	OOE
7	Soriano, José	0.3538436
8	Fedde, Erick	0.3369309
9	Rea, Colin	0.3268954
10	Gausman, Kevin	0.3264024
11	Rodón, Carlos	0.3213091
12	Flaherty, Jack	0.3204397
13	Castillo, Luis	0.3167439
14	Warren, Will	0.2905701
15	Abbott, Andrew	0.2845366
16	Anderson, Tyler	0.269966
17	Mize, Casey	0.2693619
18	Crochet, Garrett	0.2498394
19	Woo, Bryan	0.2467084
20	Verlander, Justin	0.238044
21	Cease, Dylan	0.2303953
22	Cecconi, Slade	0.229147
23	Vásquez, Randy	0.2227305
24	Hendricks, Kyle	0.1954296
25	Baz, Shane	0.1869245
26	Imanaga, Shota	0.1845945
27	Gallen, Zac	0.1797093
28	Lugo, Seth	0.177179
29	Leiter, Jack	0.1710783
30	Pallante, Andre	0.1596817
31	Yamamoto, Yoshinobu	0.1511265
32	Martinez, Nick	0.1477152
33	Springs, Jeffrey	0.1389366
34	Parker, Mitchell	0.1263605
35	Bibee, Tanner	0.1236297
36	Suárez, Ranger	0.121881
37	Cabrera, Edward	0.1081684
38	Gore, MacKenzie	0.106761
39	Fried, Max	0.0972164
40	Quantrill, Cal	0.0931702

Rank	Name	OOE
41	Freeland, Kyle	0.0790447
42	Lord, Brad	0.0664662
43	Rasmussen, Drew	0.0612022
44	Márquez, Germán	0.0448138
45	Irvin, Jake	0.0425111
46	Bradley, Taj	0.0410583
47	Falter, Bailey	0.0357632
48	Williams, Gavin	0.0216093
49	Paddack, Chris	0.0182671
50	Holmes, Clay	0.0176843
51	Burke, Sean	0.0163219
52	Patrick, Chad	0.0136721
53	Walker, Taijuan	0.0126329
54	Peterson, David	0.0083907
55	Wacha, Michael	0.0062596
56	Buehler, Walker	0.0028001
57	Quintana, Jose	0.0011253
58	Skubal, Tarik	0.0002885
59	Littell, Zack	-0.0198946
60	Kikuchi, Yusei	-0.0258006
61	Sánchez, Cristopher	-0.0262778
62	Mikolas, Miles	-0.0613987
63	Kochanowicz, Jack	-0.0638878
64	Liberatore, Matthew	-0.0695853
65	Heaney, Andrew	-0.0774083
66	Gray, Sonny	-0.0979406
67	Lodolo, Nick	-0.0982161
68	Valdez, Framber	-0.0988107
69	Giolito, Lucas	-0.1040531
70	Ober, Bailey	-0.1044305
71	Nelson, Ryne	-0.1113109
72	Kelly, Merrill	-0.1160398
73	Skenes, Paul	-0.116477
74	Martin, Davis	-0.1241761

Rank	Name	OOE
75	Bassitt, Chris	-0.1250765
76	Brown, Hunter	-0.1264336
77	Pivetta, Nick	-0.1398944
78	Taillon, Jameson	-0.1399156
79	Rodriguez, Eduardo	-0.1470984
80	Wheeler, Zack	-0.1476579
81	Webb, Logan	-0.1553434
82	Ryan, Joe	-0.1558274
83	Alcantara, Sandy	-0.1675833
84	Elder, Bryce	-0.1873644
85	Peralta, Freddy	-0.1948815
86	Singer, Brady	-0.2084301
87	Bello, Brayan	-0.2099877
88	Corbin, Patrick	-0.2121214
89	Kirby, George	-0.2134233
90	deGrom, Jacob	-0.2205878
91	Houser, Adrian	-0.2341712
92	Berríos, José	-0.2367903
93	Severino, Luis	-0.2572223
94	Morton, Charlie	-0.2626075
95	Keller, Mitch	-0.2888137
96	Luzardo, Jesús	-0.2901608
97	Sears, JP	-0.3380123
98	Smith, Shane	-0.3719033
99	Cameron, Noah	-0.3722444
100	Pfaadt, Brandon	-0.3753321
101	May, Dustin	-0.39689
102	Sugano, Tomoyuki	-0.4607614
103	Strider, Spencer	-0.4991508
104	Lorenzen, Michael	-0.522287
105	Gilbert, Logan	-0.5327203
106	Sale, Chris	-0.5572051
107	Senzatela, Antonio	-0.6827681

Note: OOE scores are measured in standard deviations

D. SkwOBA formulas, 2015-2018, 2021-2025

2015: SkwOBA = (.6229003*z_poorlytopped) + (.676829*z_poorlyunder) + (.6685517*z_kbb) + (-.3207057*z_barrel)

2016: SkwOBA = (.6883637*z_poorlytopped) + (.7577937*z_poorlyunder) + (.64867*z_kbb) + (-.3940266*z_barrel)

2017: SkwOBA = (.6448151*z_poorlytopped) + (.664817*z_poorlyunder) + (.7252123*z_kbb) + (-.3684268*z_barrel)

2018: SkwOBA = (.5239886*z_poorlytopped) + (.5906002*z_poorlyunder) + (.740456*z_kbb) + (-.3859837*z_barrel)

2021: SkwOBA = (.5706632*z_poorlytopped) + (.6113039*z_poorlyunder) + (.6606267*z_kbb) + (-.3784185*z_barrel)

2022: SkwOBA = (.6437686*z_poorlytopped) + (.6813552*z_poorlyunder) + (.6852756*z_kbb) + (-.2971638*z_barrel)

2023: SkwOBA = (.5870997*z_poorlytopped) + (.6687938*z_poorlyunder) + (.7730293*z_kbb) + (-.3480772*z_barrel)

2024: SkwOBA = (.6550074*z_poorlytopped) + (.7776122*z_poorlyunder) + (.7400951*z_kbb) + (-.2803665*z_barrel)

2025: SkwOBA = (.5962999*z_poorlytopped) + (.6124874*z_poorlyunder) + (.7026914*z_kbb) + (-.2830242*z_barrel)

Note: 2019 season excluded because it would not provide significant information on predictive value of SkwOBA due to COVID-shortened 2020 season.

E. Complete Code and Regression Outputs:

Note: a line of +++ indicates a change in the do file or dataset being used

i) Development of SkwOBA metric using 2025 data

```
generate kbb = k_percent - bb_percent
reg xwoba kbb
```

Image E1

```
. reg xwoba kbb

      Source |       SS           df       MS      Number of obs   =       107
-------------+----------------------------------   F(1, 105)       =    274.87
       Model |  .06181894         1  .06181894     Prob > F        =    0.0000
    Residual |  .023615027       105  .000224905   R-squared       =    0.7236
-------------+----------------------------------   Adj R-squared   =    0.7210
       Total |  .085433967       106  .000805981   Root MSE        =      .015

------------------------------------------------------------------------------
       xwoba | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
         kbb |  -.0045639   .0002753   -16.58   0.000    -.0051097   -.0040181
       _cons |   .3811649   .0041318    92.25   0.000     .3729723    .3893575
------------------------------------------------------------------------------
```

```
egen z_kbb = std(kbb)
egen z_poorlytopped = std(poorlytopped_percent)
egen z_poorlyunder = std(poorlyunder_percent)
replace z_xwoba = -1*z_xwoba
reg z_xwoba z_poorlytopped z_poorlyunder z_kbb z_barrel
```

Image E2

```
. reg z_xwoba z_poorlytopped z_poorlyunder z_kbb z_barrel

      Source |       SS           df       MS      Number of obs   =       107
-------------+----------------------------------   F(4, 102)       =    373.14
       Model |  99.2195193         4  24.8048798   Prob > F        =    0.0000
    Residual |  6.78048005       102  .066475295   R-squared       =    0.9360
-------------+----------------------------------   Adj R-squared   =    0.9335
       Total |  105.999999       106  .999999994   Root MSE        =    .25783

------------------------------------------------------------------------------
     z_xwoba | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
z_poorlytop~d |   .5962999   .0622477     9.58   0.000     .4728318     .719768
z_poorlyunder |   .6124874   .0568723    10.77   0.000     .4996814    .7252934
       z_kbb |   .7026914   .0265695    26.45   0.000     .6499909    .7553919
    z_barrel |  -.2830242   .0318111    -8.90   0.000    -.3461213   -.2199271
       _cons |  -2.53e-09   .0249252    -0.00   1.000     -.049439     .049439
------------------------------------------------------------------------------
```

```
generate skwoba = ((.5962999*z_poorlytopped) + (.6124874 *z_poorlyunder) +
(.7026914*z_kbb) + (-.2830242 *z_barrel))
replace skwoba = -1*skwoba
egen z_skwoba = std(skwoba)
```

++

ii) Development of OOE and initial comparisons of predictive power between 2024 and 2025 seasons

```
keep if year==2024 | year==2025
bysort player_id year: keep if _n==1
```

```
bysort player_id: egen has2024 = max(year==2024)
bysort player_id: egen has2025 = max(year==2025)
keep if has2024 & has2025
drop has2024 has2025
generate kbb = k_percent - bb_percent
egen z_kbb = std(kbb) if year==2024
egen z_poorlytopped = std(poorlytopped_percent) if year==2024
egen z_poorlyunder = std(poorlyunder_percent) if year==2024
egen z_barrel = std(barrel_batted_rate) if year==2024
egen z_xwoba = std(xwoba) if year==2024
replace z_xwoba = -1*z_xwoba if year==2024
reg z_xwoba z_poorlytopped z_poorlyunder z_kbb z_barrel if year==2024
```

Image E3

```
.
. reg z_xwoba z_poorlytopped z_poorlyunder z_kbb z_barrel if year==2024

      Source |       SS           df       MS      Number of obs   =        59
-------------+----------------------------------   F(4, 54)        =    173.54
       Model |  53.8136582         4  13.4534146   Prob > F        =    0.0000
    Residual |  4.18634122        54  .077524837   R-squared       =    0.9278
-------------+----------------------------------   Adj R-squared   =    0.9225
       Total |  57.9999994        58   .99999999   Root MSE        =    .27843

------------------------------------------------------------------------------
     z_xwoba | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
z_poorlytop~d |    .685201   .0886811     7.73   0.000     .5074061    .8629959
z_poorlyunder |   .8381243   .0835778    10.03   0.000     .6705609    1.005688
       z_kbb |   .7680729   .0370413    20.74   0.000     .6938097    .8423362
    z_barrel |  -.3336049   .0457274    -7.30   0.000    -.4252827    -.241927
       _cons |   2.02e-09   .0362489     0.00   1.000    -.0726746    .0726746
------------------------------------------------------------------------------
```

```
generate skwoba = ((.6550074*z_poorlytopped) + (.7776122*z_poorlyunder) +
(.7400951*z_kbb) + (-.2803665*z_barrel)) if year==2024
egen z_skwoba = std(skwoba) if year==2024
generate ooe = z_xwoba - z_skwoba if year==2024
egen z_ooe = std(ooa) if year==2024
keep player_id last_namefirst_name year xwoba skwoba ooe
reshape wide xwoba skwoba ooe, i(player_id last_namefirst_name) j(year)
generate xwoba_diff = (xwoba2025 - xwoba2024)
egen z_xwoba_diff = std(xwoba_diff)
egen z_skwoba2024 = std(skwoba2024)
reg xwoba2025 ooe2024
```

Image E4

```
reg xwoba2025 ooe2024

      Source |       SS           df       MS      Number of obs   =        59
-------------+----------------------------------   F(1, 57)        =      0.01
       Model |  5.7952e-06         1  5.7952e-06   Prob > F        =    0.9318
    Residual |  .044758752        57  .000785241   R-squared       =    0.0001
-------------+----------------------------------   Adj R-squared   =   -0.0174
       Total |  .044764547        58  .000771803   Root MSE        =    .02802

------------------------------------------------------------------------------
   xwoba2025 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
     ooe2024 |   .0011567   .0134643     0.09   0.932    -.0258051    .0281185
       _cons |   .3105593   .0036482    85.13   0.000      .303254    .3178647
------------------------------------------------------------------------------
```

```
reg xwoba_diff ooe2024
```

Image E5

```
. reg xwoba_diff ooe2024

      Source |       SS           df       MS      Number of obs   =        59
-------------+----------------------------------   F(1, 57)        =      1.62
       Model |  .000738686         1  .000738686   Prob > F        =    0.2077
    Residual |  .025925249        57  .000454829   R-squared       =    0.0277
-------------+----------------------------------   Adj R-squared   =    0.0106
       Total |  .026663935        58  .000459723   Root MSE        =    .02133

------------------------------------------------------------------------------
  xwoba_diff | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
     ooe2024 |  -.0130591   .0102472    -1.27   0.208    -.0335788    .0074607
       _cons |  -.0020339   .0027765    -0.73   0.467    -.0075937     .003526
------------------------------------------------------------------------------
```

++

iii) Comparisons of SkwOBA vs. lagged xwOBA predictive power on next-year xwOBA (2015-2025)

Where X is the predictor (previous) year and Y is the outcome (following) year

Note: a1, a2, a3, and a4 are the coefficients resulting from regression (1)

```
generate kbb = k_percent - bb_percent if year == X
egen z_kbb = std(kbb) if year == X
egen z_poorlytopped = std(poorlytopped_percent) if year == X
egen z_poorlyunder = std(poorlyunder_percent) if year == X
egen z_barrel = std(barrel_batted_rate) if year == X
egen z_xwoba = std(xwoba) if year == X
replace z_xwoba = -1*z_xwoba if year == X
(1) reg z_xwoba z_poorlytopped z_poorlyunder z_kbb z_barrel if year == X
generate skwoba = ((a1*z_poorlytopped) + (a2*z_poorlyunder) + (a3*z_kbb) +
(a4*z_barrel)) if year == X
generate xwobaX = xwoba if year == X
```

```
generate xwobaY = xwoba if year == Y
bysort player_id: egen n_years = count(year)
drop if n_years < 2
drop n_years
keep last_namefirst_name year xwoba skwoba
reshape wide xwoba skwoba, i(last_namefirst_name) j(year)
reg xwobaY skwobaX
reg xwobaY xwobaX
```

Regression results are as follows:

Image E6

```
. reg xwoba2016 skwoba2015

      Source |       SS           df       MS      Number of obs   =        93
-------------+----------------------------------   F(1, 91)        =     50.93
       Model |  .023779274         1  .023779274   Prob > F        =    0.0000
    Residual |  .042488008        91  .000466901   R-squared       =    0.3588
-------------+----------------------------------   Adj R-squared   =    0.3518
       Total |  .066267282        92  .000720297   Root MSE        =    .02161

------------------------------------------------------------------------------
   xwoba2016 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
  skwoba2015 |  -.0178383   .0024996    -7.14   0.000    -.0228034   -.0128732
       _cons |   .3210454   .0023239   138.15   0.000     .3164292    .3256616
------------------------------------------------------------------------------
```

Image E7

```
reg xwoba2016 xwoba2015

      Source |       SS           df       MS      Number of obs   =        93
-------------+----------------------------------   F(1, 91)        =     37.86
       Model |  .019468025         1  .019468025   Prob > F        =    0.0000
    Residual |  .046799257        91  .000514278   R-squared       =    0.2938
-------------+----------------------------------   Adj R-squared   =    0.2860
       Total |  .066267282        92  .000720297   Root MSE        =    .02268

------------------------------------------------------------------------------
   xwoba2016 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
   xwoba2015 |   .5797679   .0942306     6.15   0.000     .3925904    .7669454
       _cons |   .1403957   .0287425     4.88   0.000     .0833024    .1974891
------------------------------------------------------------------------------
```

++

Image E8

```
. reg xwoba2017 skwoba2016

      Source |       SS           df       MS      Number of obs   =        92
-------------+----------------------------------   F(1, 90)        =     44.01
       Model |  .025896512         1  .025896512   Prob > F        =    0.0000
    Residual |  .052952647        90  .000588363   R-squared       =    0.3284
-------------+----------------------------------   Adj R-squared   =    0.3210
       Total |  .078849159        91  .000866474   Root MSE        =    .02426

------------------------------------------------------------------------------
   xwoba2017 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
  skwoba2016 |   -.018381   .0027706    -6.63   0.000    -.0238853   -.0128768
       _cons |   .3202022   .0025649   124.84   0.000     .3151065    .3252979
------------------------------------------------------------------------------
```

Image E9

```
. reg xwoba2017 xwoba2016

      Source |       SS           df       MS      Number of obs   =        92
-------------+----------------------------------   F(1, 90)        =     46.81
       Model |  .026977166         1  .026977166   Prob > F        =    0.0000
    Residual |  .051871993        90  .000576355   R-squared       =    0.3421
-------------+----------------------------------   Adj R-squared   =    0.3348
       Total |  .078849159        91  .000866474   Root MSE        =    .02401

------------------------------------------------------------------------------
   xwoba2017 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
   xwoba2016 |   .6831379   .0998517     6.84   0.000     .4847651    .8815107
       _cons |   .1056082   .0310518     3.40   0.001     .0439184    .1672981
------------------------------------------------------------------------------
```

++

Image E10

```
. reg xwoba2018 skwoba2017

      Source |       SS           df       MS      Number of obs   =        85
-------------+----------------------------------   F(1, 83)        =     31.98
       Model |  .020841286         1  .020841286   Prob > F        =    0.0000
    Residual |  .054097826        83  .000651781   R-squared       =    0.2781
-------------+----------------------------------   Adj R-squared   =    0.2694
       Total |  .074939112        84  .000892132   Root MSE        =    .02553

------------------------------------------------------------------------------
   xwoba2018 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
  skwoba2017 |  -.0160971   .0028467    -5.65   0.000     -.021759   -.0104352
       _cons |   .3123939   .0028552   109.41   0.000      .306715    .3180728
```

Image E11

```
. reg xwoba2018 xwoba2017

      Source |       SS           df       MS      Number of obs   =        85
-------------+----------------------------------   F(1, 83)        =     25.23
       Model |  .017469022         1  .017469022   Prob > F        =    0.0000
    Residual |  .057470091        83  .000692411   R-squared       =    0.2331
-------------+----------------------------------   Adj R-squared   =    0.2239
       Total |  .074939112        84  .000892132   Root MSE        =    .02631

------------------------------------------------------------------------------
   xwoba2018 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
   xwoba2017 |   .5283186   .1051825     5.02   0.000     .3191148    .7375223
       _cons |   .1439529   .0328755     4.38   0.000     .0785648    .2093409
------------------------------------------------------------------------------
```

++

Image E12

```
. reg xwoba2019 skwoba2018

      Source |       SS           df       MS      Number of obs   =        83
-------------+----------------------------------   F(1, 81)        =     39.12
       Model |  .020919795         1  .020919795   Prob > F        =    0.0000
    Residual |  .043320085        81  .000534816   R-squared       =    0.3257
-------------+----------------------------------   Adj R-squared   =    0.3173
       Total |   .06423988        82  .000783413   Root MSE        =    .02313

------------------------------------------------------------------------------
   xwoba2019 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
  skwoba2018 |   -.016261      .0026    -6.25   0.000    -.0214341   -.0110878
       _cons |   .3141039   .0026067   120.50   0.000     .3089174    .3192903
------------------------------------------------------------------------------
```

Image E13

```
. reg xwoba2019 xwoba2018

      Source |       SS           df       MS      Number of obs   =        83
-------------+----------------------------------   F(1, 81)        =     38.57
       Model |  .020723727         1  .020723727   Prob > F        =    0.0000
    Residual |  .043516153        81  .000537236   R-squared       =    0.3226
-------------+----------------------------------   Adj R-squared   =    0.3142
       Total |   .06423988        82  .000783413   Root MSE        =    .02318

------------------------------------------------------------------------------
   xwoba2019 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
   xwoba2018 |   .5771147   .0929204     6.21   0.000     .3922323    .7619971
       _cons |   .1343637   .0284569     4.72   0.000     .0777434     .190984
------------------------------------------------------------------------------
```

++

Image E14

```
. reg xwoba2022 skwoba2021

      Source |       SS           df       MS      Number of obs   =        89
-------------+----------------------------------   F(1, 87)        =     33.64
       Model |  .021687669         1  .021687669   Prob > F        =    0.0000
    Residual |  .056088312        87  .000644693   R-squared       =    0.2788
-------------+----------------------------------   Adj R-squared   =    0.2706
       Total |  .077775981        88  .000883818   Root MSE        =    .02539

------------------------------------------------------------------------------
   xwoba2022 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
  skwoba2021 |  -.0160992   .0027757    -5.80   0.000    -.0216162   -.0105822
       _cons |   .3134046    .002731   114.76   0.000     .3079765    .3188327
------------------------------------------------------------------------------
```

Image E15

```
reg xwoba2022 xwoba2021

      Source |       SS           df       MS      Number of obs   =        89
-------------+----------------------------------   F(1, 87)        =     33.13
       Model |  .021449318         1  .021449318   Prob > F        =    0.0000
    Residual |  .056326663        87  .000647433   R-squared       =    0.2758
-------------+----------------------------------   Adj R-squared   =    0.2675
       Total |  .077775981        88  .000883818   Root MSE        =    .02544

------------------------------------------------------------------------------
   xwoba2022 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
   xwoba2021 |   .5447197   .0946376     5.76   0.000     .3566172    .7328222
       _cons |   .1421436   .0294116     4.83   0.000     .0836849    .2006024
------------------------------------------------------------------------------
```

++

Image E16

```
. reg xwoba2023 skwoba2022

      Source |       SS           df       MS      Number of obs   =        78
-------------+----------------------------------   F(1, 76)        =     21.49
       Model |  .010167879         1  .010167879   Prob > F        =    0.0000
    Residual |  .035957604        76  .000473126   R-squared       =    0.2204
-------------+----------------------------------   Adj R-squared   =    0.2102
       Total |  .046125483        77  .000599032   Root MSE        =    .02175

------------------------------------------------------------------------------
   xwoba2023 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
  skwoba2022 |   -.012058   .0026011    -4.64   0.000    -.0172385   -.0068776
       _cons |   .3229203   .0025328   127.49   0.000     .3178757    .3279649
```

Image E17

```
. reg xwoba2023 xwoba2022

      Source |       SS           df       MS      Number of obs   =        78
-------------+----------------------------------   F(1, 76)        =     24.42
       Model |  .011215835         1  .011215835   Prob > F        =    0.0000
    Residual |  .034909648        76  .000459337   R-squared       =    0.2432
-------------+----------------------------------   Adj R-squared   =    0.2332
       Total |  .046125483        77  .000599032   Root MSE        =    .02143

------------------------------------------------------------------------------
   xwoba2023 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
   xwoba2022 |   .4338436   .0877977     4.94   0.000     .2589793     .608708
       _cons |    .188113   .0268365     7.01   0.000     .1346636    .2415625
------------------------------------------------------------------------------
```

++

Image E18

```
. reg xwoba2024 skwoba2023

      Source |       SS           df       MS      Number of obs   =        74
-------------+----------------------------------   F(1, 72)        =     38.53
       Model |  .016225278         1  .016225278   Prob > F        =    0.0000
    Residual |  .030316939        72  .000421069   R-squared       =    0.3486
-------------+----------------------------------   Adj R-squared   =    0.3396
       Total |  .046542217        73  .000637565   Root MSE        =    .02052

------------------------------------------------------------------------------
   xwoba2024 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
  skwoba2023 |  -.0163654   .0026364    -6.21   0.000    -.0216209   -.0111099
       _cons |    .314902   .0024122   130.55   0.000     .3100934    .3197107
```

Image E19

. reg xwoba2024 xwoba2023

Source	SS	df	MS			
Model	.016815635	1	.016815635	Number of obs	=	74
Residual	.029726581	72	.000412869	F(1, 72)	=	40.73
				Prob > F	=	0.0000
				R-squared	=	0.3613
				Adj R-squared	=	0.3524
Total	.046542217	73	.000637565	Root MSE	=	.02032

xwoba2024	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
xwoba2023	.6719339	.1052873	6.38	0.000	.4620474	.8818203
_cons	.0997362	.0334496	2.98	0.004	.0330556	.1664168

Image E19

. reg xwoba2025 skwoba2024

Source	SS	df	MS			
Model	.019944493	1	.019944493	Number of obs	=	59
Residual	.024820054	57	.00043544	F(1, 57)	=	45.80
				Prob > F	=	0.0000
				R-squared	=	0.4455
				Adj R-squared	=	0.4358
Total	.044764547	58	.000771803	Root MSE	=	.02087

xwoba2025	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
skwoba2024	-.0203557	.0030077	-6.77	0.000	-.0263786	-.0143329
_cons	.3105593	.0027167	114.32	0.000	.3051193	.3159994

Image E20

. reg xwoba2025 xwoba2024

Source	SS	df	MS			
Model	.019759112	1	.019759112	Number of obs	=	59
Residual	.025005435	57	.000438692	F(1, 57)	=	45.04
				Prob > F	=	0.0000
				R-squared	=	0.4414
				Adj R-squared	=	0.4316
Total	.044764547	58	.000771803	Root MSE	=	.02094

xwoba2025	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
xwoba2024	.7753639	.1155318	6.71	0.000	.5440154	1.006712
_cons	.0713399	.0357487	2.00	0.051	-.0002456	.1429253

Image E21

```
. reg xwoba2025 skwoba2024 xwoba2024

      Source |       SS           df       MS      Number of obs   =        59
-------------+----------------------------------   F(2, 56)        =     23.10
       Model |  .020235174         2  .010117587   Prob > F        =    0.0000
    Residual |  .024529372        56  .000438025   R-squared       =    0.4520
-------------+----------------------------------   Adj R-squared   =    0.4325
       Total |  .044764547        58  .000771803   Root MSE        =    .02093

------------------------------------------------------------------------------
   xwoba2025 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
  skwoba2024 |  -.0116171   .0111433    -1.04   0.302    -.0339398    .0107056
   xwoba2024 |   .3473917   .4264421     0.81   0.419    -.5068745    1.201658
       _cons |   .2033802   .1315964     1.55   0.128    -.0602392    .4669995
```

Works Cited

"Baseball Savant: Statcast, Trending MLB Players and Visualizations." *Baseballsavant.Com*, baseballsavant.mlb.com/.

"MLB Stats, Scores, History, & Records." *Baseball Reference*, www.baseball-reference.com/.