Correcting bWAR to Account for the Relationship Between Starting Pitcher Skill and Length of Start

Milutin Gjaja
milutingjaja@gmail.com

March 21, 2025

Abstract

I use MLB pitching data from 2022 to 2024 and find a correlation between a starting pitcher's skill, as determined by bWAR, and the average number of innings he pitches per start. Based on this relationship, I claim that bWAR misvalues starting pitchers by incorrectly assuming the replacement-level player will last as long as the starting pitcher. I propose a correction to the bWAR formula and compare the resulting values to those yielded by a similar correction to fWAR and discuss implications of the result.

1 Introduction

WAR, or Wins Above Replacement, is a widely-used statistic in baseball that summarizes a player's total contribution on the field by a single number. The key concept behind the statistic is that of a "replacement-level player". Replacement-level players are players who are good enough to excel in the minors, but not good enough to get a consistent spot in a Major League team. They are assumed to be widely available and readily signed for cheap. Because of their availability and talent level, they are a good baseline against which MLB players can be measured. In short, WAR seeks to answer the following question: "If we lost a player and had to replace him with the cheapest, most readily available replacement, how much value would we lose?" WAR expresses this value in wins, usually on a per-season basis.

Although WAR allows comparisons across different positions, leagues, and years, it is far from exact. To begin with, there is no standard formula for WAR nor consensus for how to calculate it. Instead, there are numerous variants, the three most common of which are fWAR, bWAR, and WARP, developed by the publications FanGraphs, Baseball Reference, and Baseball Prospectus, respectively. These variants disagree often and considerably in their approach and as a result, a player's WAR changes depending on the variant used[1]. Within the variants themselves, WAR is acknowledged to have a particularly wide margin of error. Baseball Reference cautions to "not take any full-season difference between two players of less than one to two wins to be definitive"[2], while FanGraphs states that "WAR is not meant to be a perfectly precise indicator of a player's contribution, but rather an estimate of their value to date"[3].

This paper focuses on bWAR for starting pitchers. One of the fundamental assumptions bWAR makes is that there is a 1-for-1 substitution in game time between the starting pitcher and his replacement. I posit that this assumption is false based on the hypothesis that the better the pitcher, the more outs he pitches per start. To test this hypothesis, I study the correlation between a pitcher's skill, as defined by bWAR, and the average length of start. I then propose a modification to bWAR that takes this relationship into account and discuss its consequences. A previous paper discusses a similar issue in the fWAR formula[4]. For an up-to-date version of both papers and their associated materials, see https://github.com/milutin-gjaja/bWAR-paper.

[1]For an extreme example, see Patrick Corbin, who in 2022 had an fWAR of 0.7, a WARP of -1.6, and a bWAR of -2.3.
[2]Sports Reference LLC, "Baseball-Reference.com WAR Explained"
[3]Slowinsky, 2010
[4]Gjaja, 2025

1.1 Explanation of the bWAR formula for pitchers

Broadly speaking, the bWAR formula for pitchers can be broken down into three steps. The first calculates the player's value in wins above the average, and in doing so creates runs_above_avg, bWAR's metric of pitching skill. The next step calculates the average player's value in wins above the replacement. The final step is a simple addition of these values to get the difference in wins between the player and the replacement, with a few slight adjustments.

As with FanGraphs and fWAR[5], Baseball Reference provides a webpage that breaks down the formula components, as well as a helpful .csv file containing all component values going back to 1871[6]. Unfortunately, the online explanation is often ambiguous, lacks many explicit formulas, skips or implies multiple steps, and gives multiple definitions for the same variable. As a result, most of the formulas in this paper are the result of reading the explanation and analyzing the variables to match values in the .csv file. Since the file values match those on the Baseball Reference player pages, I take them to be the correct, official values. Moreover, to avoid confusion, all variable names used in this paper are those in the .csv file, even when those names are complicated or misleading. For a detailed explanation of all variables names and full explicit formulas, please refer to Appendix A.

bWAR begins its calculation with the simplest possible measure of a pitcher's skill: RA, or Runs Allowed, the number of runs a pitcher has given up over the course of a season. To contextualize this number, it creates an expected RA metric, xRA_final, whose value is unique to each pitcher's situation. xRA_final takes into account factors such as which parks the pitcher played in and how good his defense was. Put simply, xRA_final is the number of runs an average pitcher would allow given the same circumstances as the pitcher in question.

$$\text{runs\_above\_avg} = \text{xRA\_final} - \text{RA}$$

The variable runs_above_avg represents the pitcher's performance below or above that of an average pitcher and is calculated as a simple difference between the expected and actual runs. Note that runs_above_avg is the number of runs a pitcher *saves* above the average; as a result, a pitcher who outperforms expectations will have a positive runs_above_avg value. A minor adjustment, runs_above_avg_adj, averages the sum of all runs_above_avg for a given year to zero.

The next step is converting the difference between pitcher and average from runs to wins. To do this, bWAR uses a formula known as PyPat, which is a modification of baseball's Pythagorean Theorem developed by sabermetrician Bill James. PyPat takes the number of runs per game a team scores (teamRpG) and the number of runs per game a team allows (oppRpG) and returns that team's win percentage. For an in-depth explanation of PyPat, please refer to Appendix A.2.

$$\text{oppRpG} = \text{teamRpG} - \frac{\text{runs\_above\_avg\_adj}}{\text{G}}$$

In the bWAR formula, teamRpG is the average number of runs scored per game by a team for a given year and league. To get oppRpG, we divide the pitcher's performance compared to average by the number of games he played and take the difference with teamRpG. Thus, oppRpG represents the number of runs per game an average team would allow, taking into account the pitcher's impact.

After plugging these values into PyPat, we get a win percentage, waa_win_perc, which represents the win percentage of an average team taking into account the pitcher's impact. Since this value is in wins per game and we want wins, the last step is to take the difference with 0.5 (since we are comparing with the average) and to scale by the number of games. The resulting value, called WAA, represents the number

[5] Slowinski, 2010

[6] Sports Reference LLC, "Baseball-Reference.com WAR Explained." The .csv file is under the "Download Our WAR Numbers Daily" heading.

of wins the pitcher is worth above the average player.

$$\text{WAA} = (\text{waa\_win\_perc} - 0.5) \times \text{G}$$

WAA gives us the pitcher's contribution above the average. To find the average pitcher's contribution above the replacement, we follow a similar process. First, we find the difference in runs between the average player and the replacement over the course of the season, denoted avg_runs_above_rep[7].

$$\text{avg\_runs\_above\_rep} = \left(\text{RpO\_replacement} - \frac{\text{teamRpG}}{\text{outs\_per\_game\_cnst}}\right) \times \text{IPouts}$$

avg_runs_above_rep is found by taking the difference between number of runs per out a replacement player allows (RpO_replacement) and the number of runs per out an average pitcher allows (found by dividing the average number of runs scored per game, teamRpG, by a constant representing the number of outs per game, outs_per_game_cnst). This is then scaled by the number of outs the pitcher played. As with runs_above_rep, avg_runs_above_rep represents the number of runs *saved* by the average over the replacement.

$$\text{oppRpG\_rep} = \text{teamRpG} + \frac{\text{avg\_runs\_above\_rep}}{\text{G}}$$

The rest of the replacement calculation runs parallel to that of the pitcher: we calculate oppRpG_rep[8], the oppRpG equivalent taking into account the replacement's performance, plug oppRpG_rep and teamRpG into PyPat, and come out the other side with the win percentage of an average team with the replacement pitcher. Taking the difference with 0.5 and scaling by games results in WAR_rep, the replacement contribution to bWAR[9].

$$\text{bWAR} = \text{WAA} + \text{WAA\_adj} + \text{WAR\_rep}$$

The final bWAR value is a simple addition of WAA and WAR_rep, plus a small adjustment factor, WAA_adj, that averages WAA to zero and takes into account leverage for reliever pitchers (see Appendix A.4.1 for further discussion of this factor). Figure 1 summarizes the main steps in the bWAR calculation.

[7] The online explanation never precisely defines the difference in runs between replacement and average, nor by extension oppRpG_rep. As a result, I've created a streamlined version of the formula using avg_runs_above_rep and outs_per_game_cnst. This method results in oppRpG_rep values which are extremely close to the official ones, differing by an average of 0.005 and by a maximum of 0.009. For an in-depth discussion of this issue, please refer to Appendix A.4.2.

[8] Note that oppRpG_rep adds the replacement contribution to teamRpG instead of subtracting it, since it is presumed that the replacement is worse than the average.

[9] The name WAR_rep is somewhat misleading, since this value represents the number of wins an average pitcher is worth over a replacement pitcher. Since the replacement level is below the average, this value will always be positive, unlike WAA which can be negative if the pitcher underperforms the average.

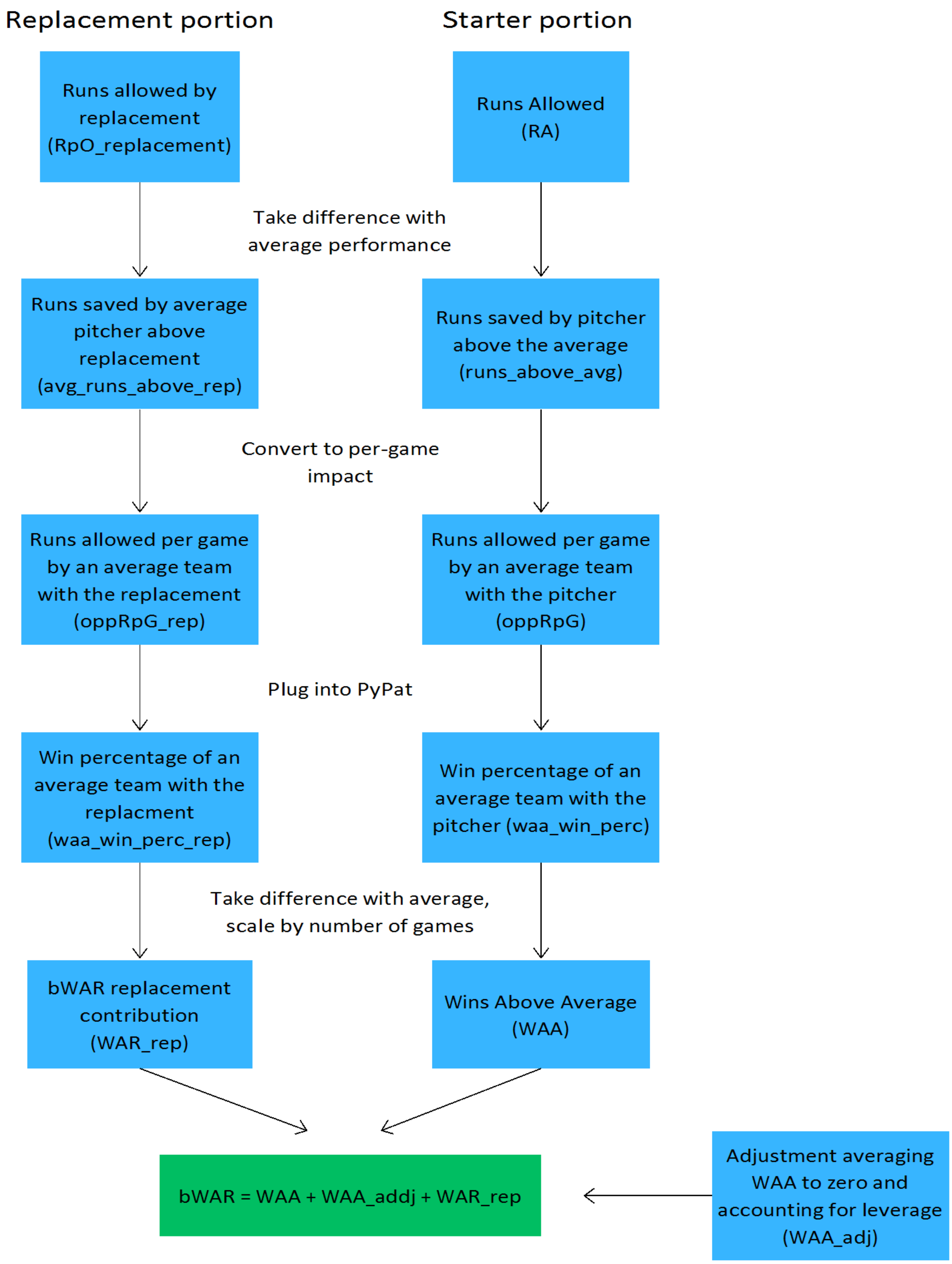


Figure 1: Summary of bWAR calculation process

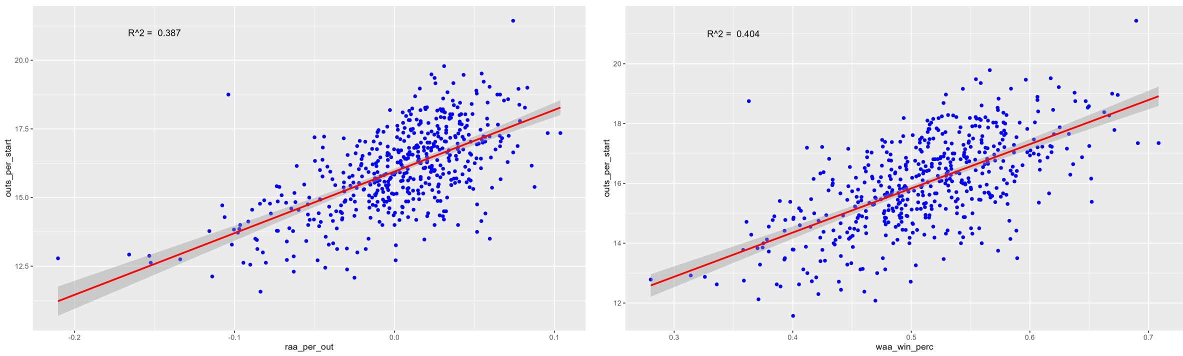
2 Methods

For this study, I use pitching data from 2022 to 2024 and set GS >= 7 and GS/G >= 0.8 to filter out relief pitchers. This gives me n = 512 observations. Almost all data is taken from the aforementioned .csv file, which is regularly updated and is available for download on the Baseball Reference website[10]. The exception are bullpen stats used to calculate the bullpen contribution; these were taken from FanGraphs[11] because it provides a more convenient breakdown by league and year than Baseball Reference.

3 Results

3.1 Relationship between pitcher skill and length of start

As discussed, the base metric bWAR uses to evaluate a pitcher's contribution is RA, which it puts into context with xRA_final, resulting in runs_above_avg. I take this to be the formula's measure of a pitcher's skill, which can be scaled to a per-out basis. A possible alternative is waa_win_perc, the win percentage of an average team with the pitcher in question. I conduct two linear regressions to study the relationship between the pitcher's skill and the average number of outs pitched in a game: outs_per_start $\sim$ raa_per_out and outs_per_start $\sim$ waa_win_perc[12]. The results are summarized in Figure 2 and Table 1.



(a) raa_per_out

(b) waa_win_perc

Figure 2: Relationship between outs_per_start and skill

Measure	Coef.	Est.	Std. Error	t-value	Res. SE	R^2	p-value
raa_per_out	Intercept	15.95	0.05	296.54	1.21	0.39	< 2.2e-16
	raa_per_out	22.44	1.25	17.94			
waa_win_perc	Intercept	8.45	0.41	20.58	1.12	0.40	< 2.2e-16
	waa_win_perc	14.77	0.80	18.57			

Table 1: Summary of relationship between outs_per_game and skill

The regressions show a clear correlation between both measures of skill and the length of a pitcher's start. R^2 values are around 0.4 and all t-stats are very high, indicating a statistically strong relationship. This evidence clearly demonstrates that the better a pitcher is, the longer his starts last. This in turn indicates a similar flaw in bWAR as in fWAR: the formula erroneously assumes a 1-for-1 substitution between starter and replacement, who does not last as long.

To illustrate why this matters, Figure 3 shows the number of runs allowed per out against the number of outs per start for all observations[13]. Values are summarized in Table 2. The lines represent the mean values of three groups: solid for all observations, dotted for elite players, dashed for replacement-level players. The replacement-level outs_per_start is the average value across all years of RpO_replacement

[10] Sports Reference LLC. "Daily Updated Pitching WAR data (in CSV)."

[11] These can be found at https://www.fangraphs.com/leaders/major-league.

[12] outs_per_start is GS/IPouts_start and raa_per_out is runs_above_avg/IPouts.

[13] I use RA instead of runs_above_avg because the replacement level and bullpen do not get individual adjustments based on the pitcher's circumstances, making a direct comparison with runs_above_avg stat difficult. ra_per_out is RA/IPouts.

and the outs_per_start value is the average across all players in the dataset with -0.5 <= bWAR <= 0.5. Elite values are averages for players in the top quartile by bWAR. The additional yellow line is the average bullpen ra_per_out across all three seasons.

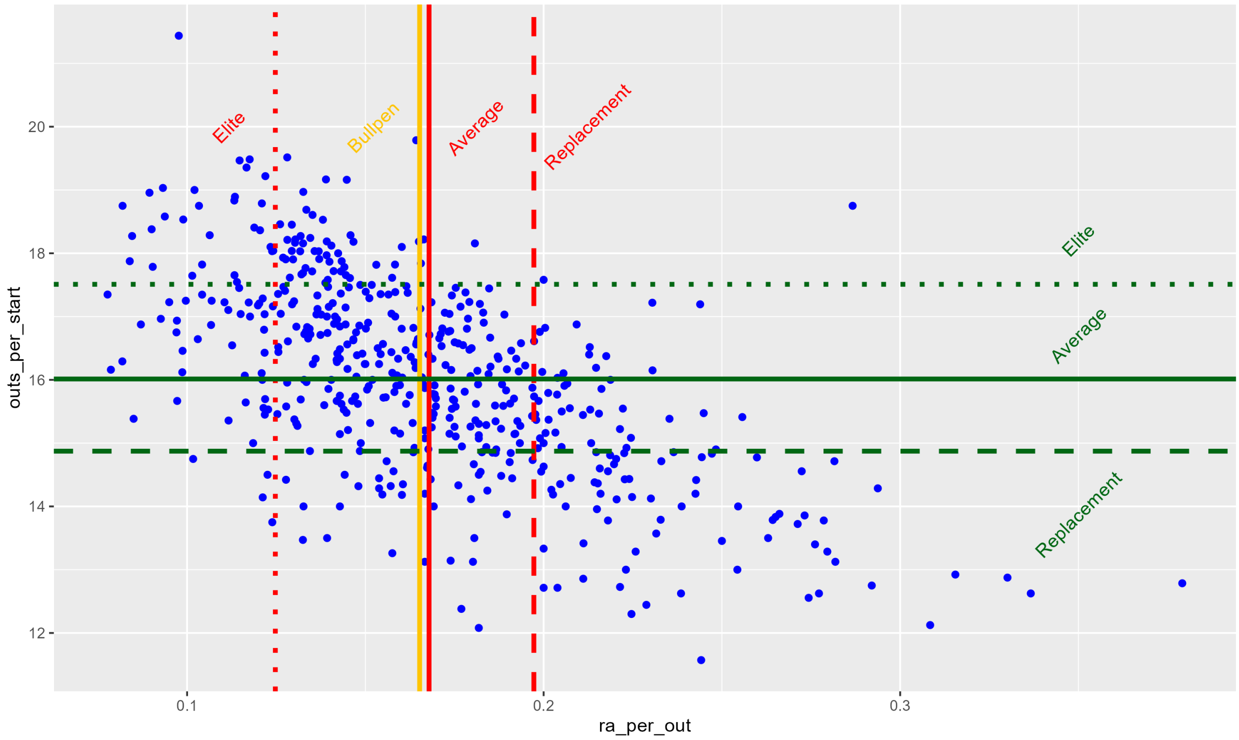


Figure 3: Relationship between outs_per_start and ra_per_out with benchmarks

Benchmark	outs_per_start	ra_per_out
Replacement-level	14.87	0.197
Average	16.01	0.168
Elite	17.51	0.125
Bullpen	NA	0.165

Table 2: Summary of benchmarks

Figure 3 shows the clear gap in both skill and length of start between elite, average, and replacement. Because of this gap, bWAR attributes an average of 1.14 and a maximum of 6.56 outs per game to the replacement when instead the bullpen would be on the mound. Over the course of a season, this adds up to an average of 33.32 and a maximum of 201.07 outs[14]. Since the bullpen is far better than the replacement in terms of skill, this results in bWAR overvaluing the contribution of starting pitchers, particularly high-end ones.

3.2 Proposed correction: new_bWAR

Unfortunately, bWAR does not present as intuitive a fix as fWAR did. Because of the use of PyPat, which only uses per-game metrics, all per-out modifications have to occur relatively early in the formula. Moreover, bWAR keeps the replacement and pitcher contributions separate until the very end. As a result, a correction similar to the one proposed to fWAR does not work for bWAR. Therefore, I propose the following correction, which I believe matches bWAR's approach most closely.

[14]Both maximum values come from Sandy Alcantara, who started 32 games and averaged a whopping 21.44 outs per start in 2022, over two full innings more than the replacement. A "normal" elite pitcher is closer to 2.64 outs per game and 77.46 outs per season above the replacement level.

$$\text{scaled\_avg\_runs\_above\_rep} = \left(\text{RpO\_replacement} - \frac{\text{teamRpG}}{\text{outs\_per\_game\_cnst}}\right)$$

$$\times\ (\text{GS} \times \text{outs\_per\_start\_rep} + \text{IPouts\_relief})$$

The first fix is to scale the replacement contribution to accurately reflect how many innings he would pitch per start. The relief innings remain untouched.

$$\text{scaled\_bp\_runs\_above\_avg} = \left(\frac{\text{teamRpG}}{\text{outs\_per\_game\_cnst}} - \text{bullpenRpO}\right)$$

$$\times\ \text{GS} \times (\text{outs\_per\_start} - \text{outs\_per\_start\_rep})$$

To calculate the bullpen contribution, we begin with bullpenRpO, the number of runs allowed per out by an average bullpen by year and by league. I use a league-wide constant for the bullpen instead of going team-by-team, as I did in my previous paper. There are two reasons for this. The first is that the other baselines in the formula, namely RpO_replacement and teamRpG, are constant by year and by league. The second is that keeping the bullpen constant allows for better forward projection[15]. The downside is that on a player-by-player basis, bWAR becomes a little less precise. Since bWAR's philosophy is to compare the pitcher to a set of average circumstances, I believe making the bullpen constant is the best way to approach a correction.

As with the starter and replacement, we take the difference with the average and we scale that by the number of outs each game that the replacement would not cover[16]. The rest is straightforward. scaled_avg_runs_above_rep takes the place of avg_runs_above_rep, while scaled_bp_runs_above_avg is used to create a third oppRpG metric, oppRpG_bp. The three oppRpG values (for starter, replacement, and bullpen) are then plugged into PyPat, resulting in three separate win percentages. The bullpen's wins above average, WAA_bp, is found by subtracting 0.5 from the corresponding PyPat win percentage and scaling by the number of games. The final correction formula is as follows:

$$\text{new\_bWAR} = \text{WAA} + \text{WAA\_adj} + \text{new\_WAR\_rep} - \text{WAA\_bp}$$

Since what we're ultimately looking for is the difference in wins between the starter and the bullpen, we subtract the bullpen WAA in the final sum. If the bullpen turns out to be below average, WAA_bp will be negative, which will result in the pitcher gaining value. WAA and WAA_adj are untouched, since they only involve the starting and average pitchers. Like bWAR, the resulting new_bWAR is rounded to two decimal points. Figure 4 illustrates the main steps of the updated calculation.

[15] Thank you to Timothy Wise of the Mets for this suggestion.

[16] Because we want the number of runs *saved* by the bullpen above the average, we subtract the bullpen runs from the average runs instead of the other way around, similar to how we subtract RA from xRA_final for the starting pitcher. The reason for doing this is, unlike the replacement level, there is no guarantee that the bullpen's level will be above average. A sub-average bullpen level will result in a negative scaled_bp_runs_above_avg, and down the line, a win percentage below 0.5.

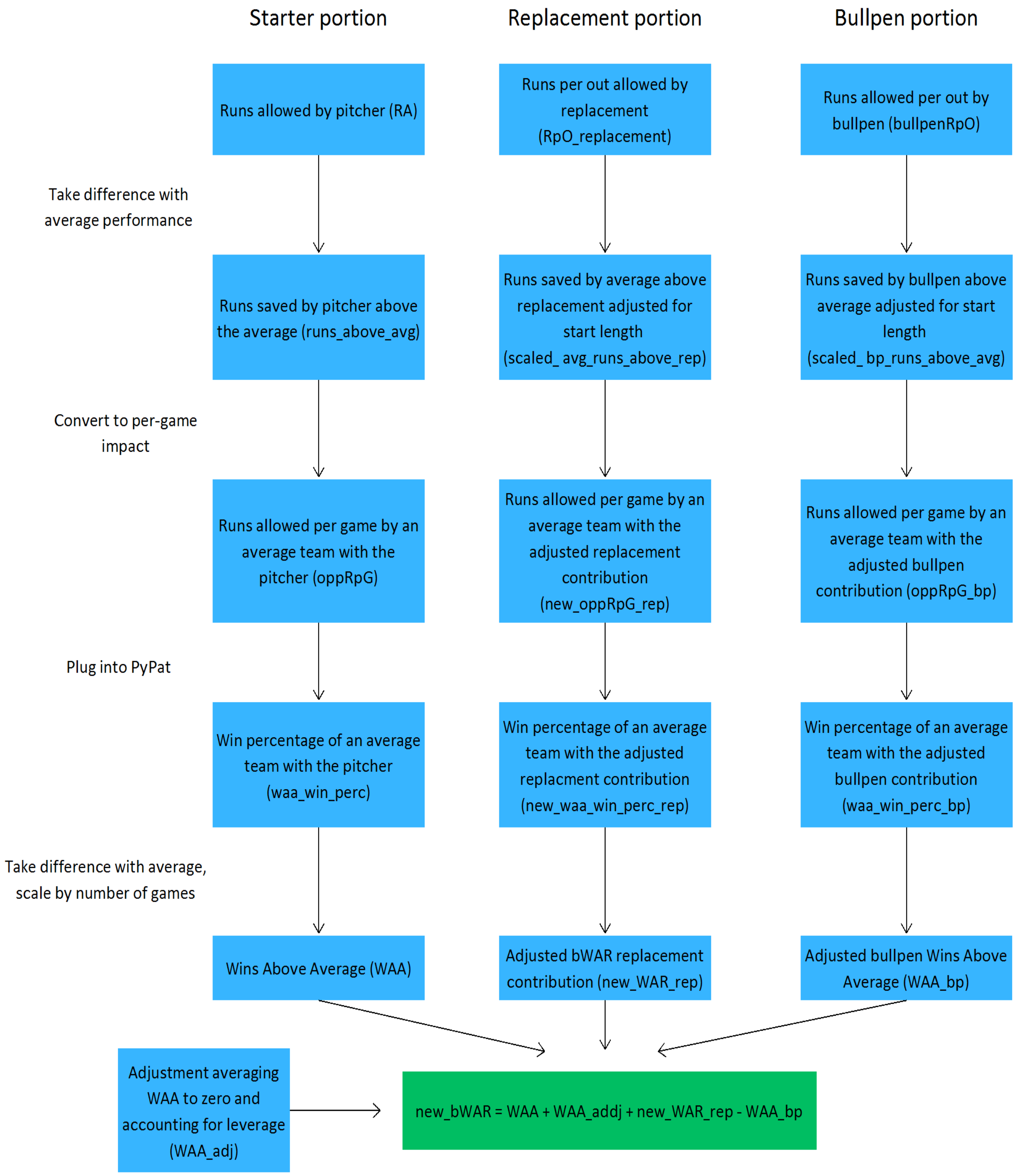


Figure 4: Summary of new_bWAR calculation process

3.3 Difference between bWAR and new_bWAR

Figure 5 shows the change between bWAR and new_bWAR by deciles of WAR. A summary of the difference is provided in Table 3, with an additional row illustrating the impact on the top 10% of pitchers by WAR. While there is a decrease in bWAR value across all observations, it averages a negligible 0.11. The change is more noticeable in elite pitchers, averaging 0.28, but remains well within the inherent uncertainty of WAR as a measure.

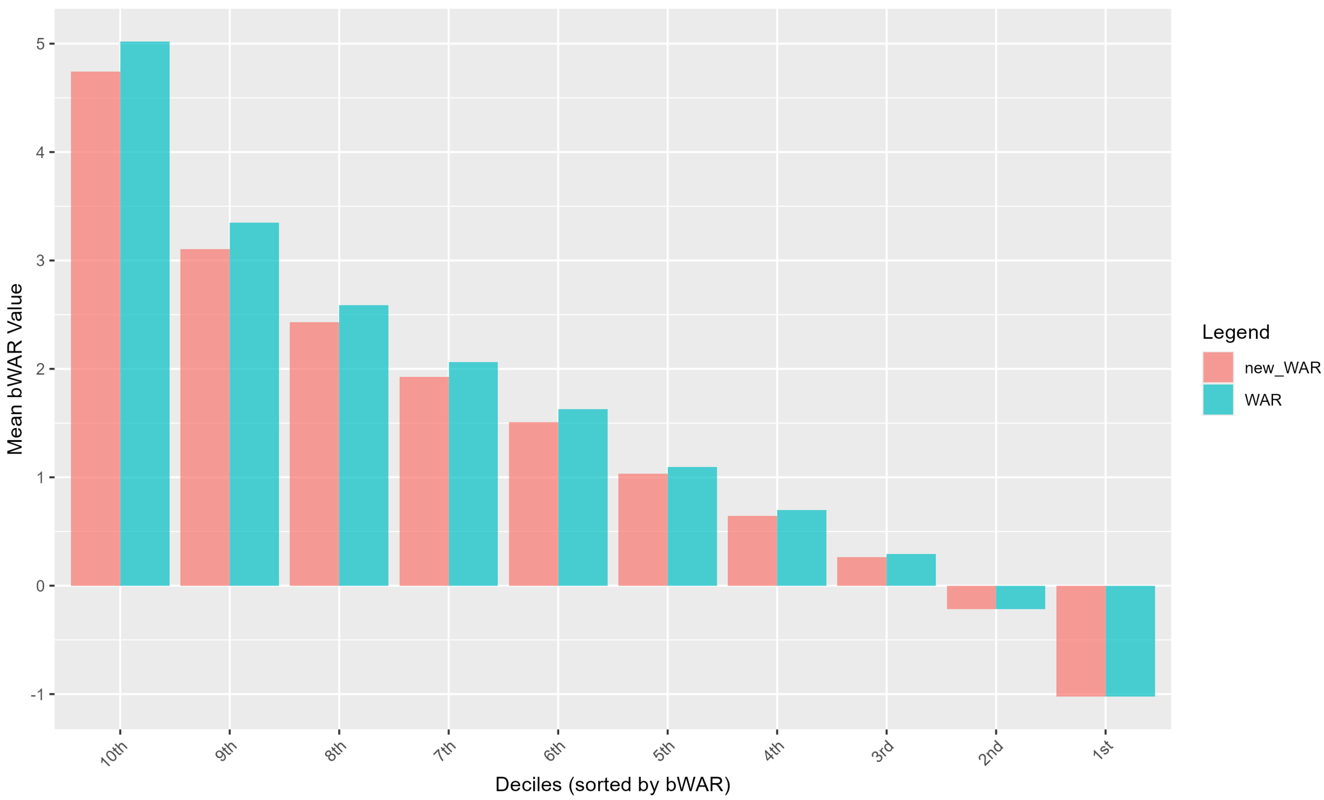


Figure 5: Comparison between bWAR and new_bWAR by bWAR deciles

Dataset	n=	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
All	512	-0.22	0.01	0.08	0.11	0.19	0.58
Top 10% of bWAR	51	-0.10	0.21	0.28	0.28	0.34	0.58

Table 3: Summary of bWAR − new_bWAR

Figure 6 shows the 10 largest changes in bWAR in the observations. As expected, these are the elite of the league: every player present pitches over an inning per game longer than the replacement and has a high runs_above_avg value. Nonetheless, the loss in bWAR remains modest. The largest drop is 0.58 for Alcantara '22, followed by 0.53 for Webb '23 and 0.50 for Valdez '22.

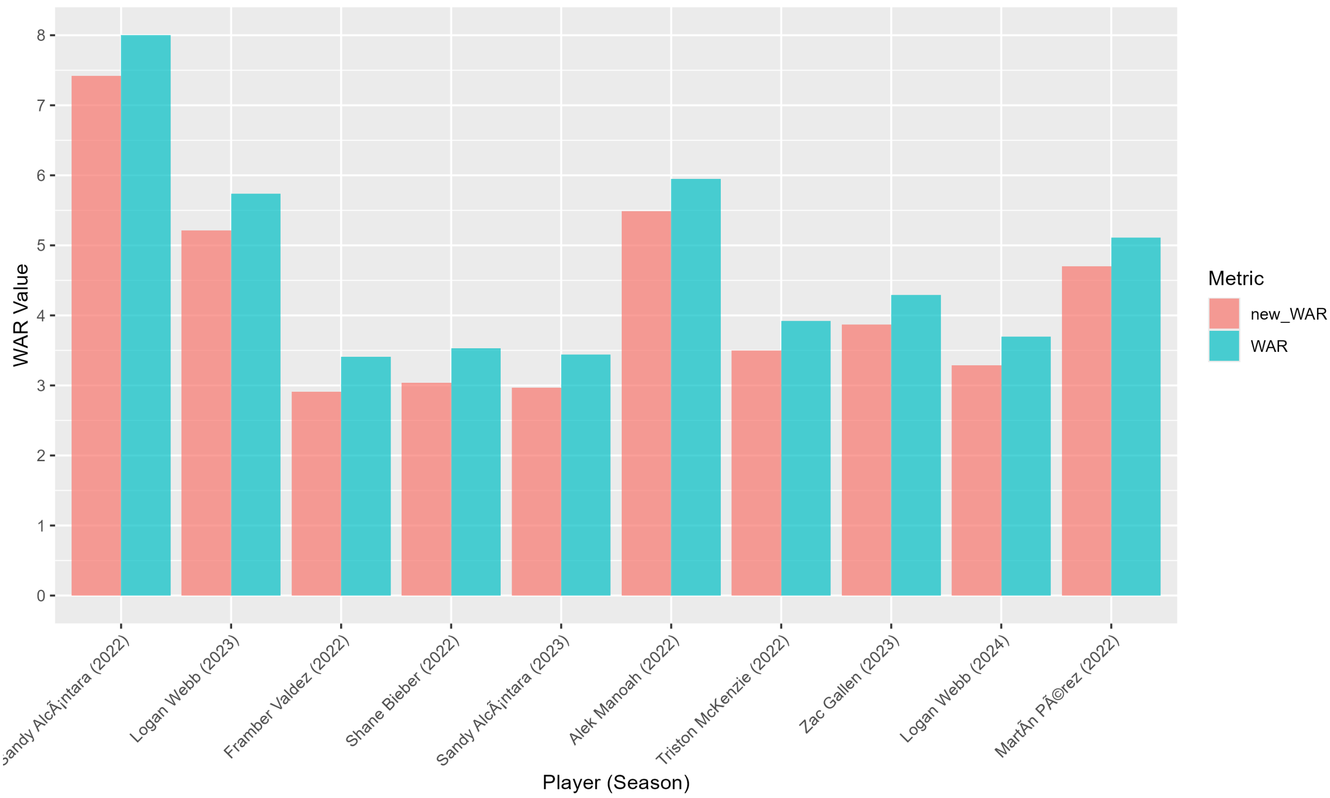


Figure 6: Ten largest losses of bWAR

4 Conclusion

Two conclusions can be made from this study. First, due to the correlation between pitcher skill and length of start, bWAR consistently overvalues starting pitchers; high-end ones are affected the most. Second, this overvaluation is small: only for the elite does the difference average over 0.2 bWAR, and for the vast majority of pitchers it is comparable to the inherent uncertainty of WAR as a metric.

It is this second conclusion that is most surprising. My proposed correction to the same flaw in fWAR resulted in significantly larger changes, as Table 4 demonstrates[17]. On the individual level, the average correction to fWAR has twice the impact the corresponding correction on bWAR[18]. This results in the total yearly change in fWAR values being nearly double that of bWAR. Elite starters remain those with the largest changes in both metrics, although the impact of that change is significantly larger for fWAR than bWAR: 35 players have their fWAR value drop by more than 0.58, the maximum change in bWAR.

Metric	Mean correction	Mean correction for top 10%	Max correction	Total yearly correction
fWAR	0.21	0.56	1.14	36.06
bWAR	0.12	0.28	0.58	21.16

Table 4: Comparison between fWAR and bWAR corrections

This discrepancy is likely due to internal choices bWAR makes with its formula, crucially the use of PyPat. PyPat uses per-game stats, meaning any per-out modifications have to be made before it. Since bWAR keeps the pitcher/average and average/replacement differences separate until the very end of the formula, the correction can only be applied to the average/replacement difference, unlike the correction to fWAR which is applied directly to the pitcher/replacement difference.

This separation is important because PyPat is a non-linear transformation. As such, the pitcher/average

[17] Table 4 uses absolute values, which is why the corresponding means are slightly higher than in Table 3.

[18] The proposed correction to fWAR took into account individual team bullpens; while implementing a standard bullpen level would certainly change individual fWAR values, it seems unlikely it would affect the averages present in the table.

and average/replacement differences are scaled differently depending on where the average lies, unlike in fWAR[19]. For elite pitchers, the pitcher/average skill gap is significantly larger than the average/replacement gap. This means the former takes an outsized importance in the final formula compared to the latter, to which the correction is applied. As a result, the corrected portion of the formula is minimized precisely for those pitchers where the gap in talent and length of start with the replacement is most significant.

Finally, bWAR confounds two different averages, further muddying the impact of the correction. The pitcher's run count is compared to the performance of an average pitcher with an identical schedule, defensive contribution, park factor, etc. The replacement's run count, however, is compared to the *league* average, which disregards the pitcher's particular circumstances. This potentially creates a gap between the two averages which remains unaccounted for throughout further calculations.

None of these choices are necessarily wrong, of course, since there is no consensus on how to calculate WAR. An in-depth study comparing the different WAR formulas and philosophies is necessary to elucidate these questions and potentially propose both a standardized method of calculation and correction.

As with the correction to fWAR, the main factor that this correction does not take into account is the effect of additional playing time on the bullpen. Should a pitcher get replaced for a full season, as his bWAR value assumes, the bullpen would be responsible for the gap between the replacement's start and the pitcher's start. This can amount to 30 or 40 outs for elite pitchers over the course of the season. There is a general consensus that increasing the bullpen's playing time decreases its effectiveness, but I am not aware of any in-depth study that quantifies this phenomenon. Even if the bullpen's level were to decrease, it would almost certainly remain above that of the replacement, and thus the gap in skill level this paper discusses would still exist.

Finally, I offer the following thoughts on what this study and the previous one on fWAR entail for pitching strategy and roster construction:

1. High-end pitchers are the only group whose loss of WAR is significant, and by extension the only group which can be confidently said to be overvalued (at least from WAR's perspective).
2. Restrictions on roster size and the necessity to have at least five pitchers in the rotation make it impractical to increase the number of relievers.
3. From the correction's perspective, the vast majority of the value gained by the starter is in the first 14.87 outs, or approximately five innings, when the replacement would be pitching.

These points suggest a path forward. Roster-wise, reduce spending on elite starters, who are overvalued, and invest instead in improving the quality (but not necessarily quantity) of the bullpen. Strategy-wise, have starters pace themselves for five innings, rather than six or seven; doing so would presumably increase their effectiveness for those innings, which is where most of their value comes from. Of course, there are far more considerations to take into account. For instance, there is almost certainly value in the additional outs afforded by elite pitchers that bWAR does not take into account, such as flexibility regarding bullpen injuries, or the playoffs, where the bullpen is already overworked and an extra inning or two can result in a crucial day of rest for a reliever. For this topic as well, further study is needed.

References

Sports Reference LLC. n.d. "Baseball-Reference.com WAR Explained." Baseball-Reference.com - Major League Statistics and Information. Accessed March 16, 2025.
https://www.baseball-reference.com/about/war_explained.shtml.

———. n.d. "Pitcher WAR Calculations and Details." Baseball-Reference.com - Major League Statistics and Information. Accessed March 16, 2025b.
https://www.baseball-reference.com/about/war_explained_pitch.shtml.

[19]The approach of applying the correction to the average/replacement difference and subtracting the bullpen/average difference can also be applied to fWAR. Unlike bWAR, the resulting values will be identical to those obtained by scaling the starter/replacement directly, since everything that follows the correction in fWAR is linear.

———. n.d. "Position Player WAR Calculations and Details." Baseball-Reference.com - Major League Statistics and Information. Accessed March 16, 2025c.
https://www.baseball-reference.com/about/war_explained_position.shtml.

———. n.d. "Baseball-Reference.com WAR Explained, Converting Runs to Wins." Baseball-Reference.com - Major League Statistics and Information. Accessed March 16, 2025b.
https://www.baseball-reference.com/about/war_explained_runs_to_wins.shtml.

———. n.d. "WAR Download Glossary and Column Headings." Baseball-Reference.com - Major League Statistics and Information. Accessed March 16, 2025e.
https://www.baseball-reference.com/about/war_explained_glossary.shtml.

———. n.d. "Daily Updated Pitching WAR data (in CSV)" Baseball-Reference.com - Major League Statistics and Information. Accessed March 16, 2025f.
https://www.baseball-reference.com/data/war_daily_pitch.txt.

———. 06/04/21. "Pythagorean Theorem of Baseball" Baseball-Reference.com - Major League Statistics and Information. Accessed March 16, 2025g.
https://www.baseball-reference.com/bullpen/Pythagorean_Theorem_of_Baseball

Slowinski, Piper. 02/15/2010. "What Is WAR? — Sabermetrics Library." Sabermetrics Library.
https://library.fangraphs.com/misc/war/.

A bWAR formula details

A.1 Variable names

G, GS: Games and Games Started. G is the number of games the pitcher played in and GS is the number of games the pitcher started. As a result, GS <= G.

RA: Runs Allowed. The number of runs (both earned and unearned) a pitcher has given up over the course of a season.

IPouts: The number of outs a pitcher has pitched. Can be divided by 3 to get the number of innings. IPouts_start and IPouts_relief are the number of outs pitched as a starter and as a reliever, respectively: IPouts = IPouts_start + IPouts_relief.

xRA: Expected Runs Allowed. See formula below. Note that this doesn't take into account context such as defensive contribution, park factor, etc.

BIP_perc: Balls in Play percentage. The percentage of balls put into play against the team over the course of a season that were allowed by the pitcher. This is used to calculate the defensive contribution.

RS_def_total: Runs Saved by the defense. From 2003 on, bWAR uses Baseball Info Solutions' Runs Saved to calculate this stat.

xRA_def_pitcher: The total defensive contribution the pitcher had over the course of the season.

xRA_sprp_adj: Stands for expected runs against starting pitcher relieving pitcher adjustment. A minor adjustment that accounts for the fact that starting pitchers have to pace themselves, whereas relievers can throw hard immediately because they're only expected to last a few outs.

xRA_extra_adj: A minor adjustment accounting for extra innings. Starting in 2023, a runner is placed at second base at the beginning of each extra inning which impacts the pitcher's run expectancy.

PF and PFF_custom: Park Factor and custom Park Factor. A stadium's park factor accounts for how hitter or pitcher friendly a park is, considering factors such as weather, stadium dimensions, and altitude. Park factors average 100. PFF_custom is a park factor tailored to the pitcher's specific schedule.

xRA_final: The expected runs allowed taking into account the adjustments listed below. In theory, the runs allowed by an average pitcher given the circumstances of the pitcher in question.

runs_above_avg: The pitcher's actual performance compared to the expected runs allowed value. Note that this is expresses the number of runs *saved* above average; a pitcher who outperforms expectations will have a positive runs_above_avg value.

runs_above_avg_adj: A minor modification so that runs_above_avg averages 0 by year and league.

teamRpG: Team Runs per Game. The average number of runs in a game, divided by 2, by year and league. Represents the average performance.

oppRpG: Opponent Runs per Game. The average number of runs per game an average team allows with the pitcher in question, by year and league. Represents the average performance taking the pitcher's contribution into account.

pyth_exponent: The exponent for the PyPat formula. See discussion of PyPat formula below.

waa_win_perc: Wins Above Average win percentage. The win percentage of an average team with the pitcher in question, as estimated by PyPat.

WAA: Wins Above Average. The pitcher's value in wins above the average over the course of a season. This value is negative for pitchers who underperform the average.

GR_leverage_index_avg: The average leverage when the pitcher enters the game in relief. For starters who do not pitch any outs in relief, this is set to 1.

WAA_adj: Wins Above Average adjustment. A minor adjustment factor taking into account leverage and a factor to average WAA to 0 by year. See Appendix X for further discussion.

outs_per_game_cnst: A by-league and by-year constant used to calculate the replacement's contribution. For an extended discussion of this and other replacement-related variables, see Appendix X.

RpO_replacement: The number of runs per out allowed by a replacement-level pitcher. Set by year and league.

runs_above_rep: The difference in runs allowed between the pitcher and the replacement. As with runs_above_avg, this represents runs *saved*, and thus will be positive for pitchers who outperform the replacement.

avg_runs_above_rep: The difference in runs allowed between the average and the replacement. Similar to runs_above_rep and runs_above_avg, represents the runs *saved*.

oppRpG_rep, pyth_exponent_rep, waa_win_perc_rep: The replacement-level counterparts to the oppRpG, pyth_exponent, and waa_win_perc. It is assumed that the replacement level is lower than the average.

WAR_rep: The replacement portion of the WAR formula. A confusingly-named variable which represents the difference in wins between an average team and an average team with a replacement-level pitcher.

bWAR: Baseball Reference's final Wins Above Replacement stat.

A.2 The use of PyPat

To convert from runs saved to wins, bWAR relies on a formula known as baseball's Pythagorean Theorem. Developed by sabermetrician Bill James, the formula estimates a team's win percentage based on the number of runs the team scores, RS, and the number of runs the team allows, RA[20].

$$\text{Win Percentage} = \frac{\text{RS}^2}{\text{RS}^2 + \text{RA}^2}$$

Further research by David Smyth, working under the pen name Patriot, created a floating exponent based on the number of runs scored in a game, both for and against. The resulting formulas, known as PyPat, are as follows[21]:

$$\text{PyPat\_exp} = \left(\frac{\text{RS+RA}}{\text{G}}\right)^{0.285}$$

$$\text{PyPat Win Percentage} = \frac{\text{RS}^{\text{PyPat\_exp}}}{\text{RS}^{\text{PyPat\_exp}} + \text{RA}^{\text{PyPat\_exp}}}$$

bWAR modifies these formulas to measure the pitcher's impact on the team's win percentage. It replaces $\frac{\text{RS}}{\text{G}}$ by teamRpG, the average runs scored by a team in a given year and league. To get the

[20] See Sports Reference LLC, "Pythagorean Theorem of Baseball" for further information on this formula.

[21] The exponent of 0.285 is sometimes changed to 0.287 or 0.283 for other uses of PyPat. bWAR uses 0.285.

equivalent of $\frac{\text{RA}}{\text{G}}$, it modifies teamRpG to account for the pitcher's impact.

$$\text{oppRpG} = \text{teamRpG} - \frac{\text{runs\_above\_avg\_adj}}{\text{G}}$$

bWAR then plugs these values into the PyPat formula to get the team's winning percentage with the pitcher's contribution.

$$\text{pyth\_exponent} = (\text{teamRpG} + \text{oppRpG})^{0.285}$$

$$\text{waa\_win\_perc} = \frac{\text{teamRpG}^{\text{pyth\_exponent}}}{\text{teamRpG}^{\text{pyth\_exponent}} + \text{oppRpG}^{\text{pyth\_exponent}}}$$

The same process is used for the replacement. Instead of oppRpG, it uses oppRpG_rep. It then calculates a new PyPat exponent, pyth_exponent_rep, and a new win percentage, waa_win_perc_rep. The proposed correction creates a third iteration of the same process for the bullpen.

A.3 Full explicit formulas

A.3.1 bWAR

$$\text{xRA} = \sum_{i=1}^{30} \text{runs\_scored\_per\_out}_i \times \text{IPouts}_i$$

$$\text{xRA\_def\_pitcher} = \text{BIP\_perc} \times \text{RS\_def\_total}$$

$$\text{PFF\_custom} = \frac{\sum_{j=1}^{30} \text{PF}_j \times \text{G}_j}{\text{G}}$$

$$\text{xRA\_final} = \frac{\text{PPF\_custom}}{100} \times (\text{xRA} - \text{xRA\_def\_pitcher} + \text{xRA\_sprp\_adj} + \text{xRA\_extras\_adj})$$

$$\text{runs\_above\_avg} = \text{xRA\_final} - \text{RA}$$

$$\text{oppRpG} = \text{teamRpG} - \frac{\text{runs\_above\_avg\_adj}}{\text{G}}$$

$$\text{pyth\_exponent} = (\text{teamRpG} + \text{oppRpG})^{0.285}$$

$$\text{waa\_win\_perc} = \frac{\text{teamRpG}^{\text{pyth\_exponent}}}{\text{teamRpG}^{\text{pyth\_exponent}} + \text{oppRpG}^{\text{pyth\_exponent}}}$$

$$\text{WAA} = (\text{waa\_win\_perc} - 0.5) \times \text{G}$$

$$\text{avg\_runs\_above\_rep} = \left(\text{RpO\_replacement} - \frac{\text{teamRpG}}{\text{outs\_per\_game\_cnst}}\right) \times \text{IPouts}$$

$$\text{oppRpG\_rep} = \text{teamRpG} + \frac{\text{avg\_runs\_above\_rep}}{\text{G}}$$

$$\text{pyth\_exponent\_rep} = (\text{teamRpG} + \text{oppRpG\_rep})^{0.285}$$

$$\text{waa\_win\_perc\_rep} = \frac{\text{teamRpG}^{\text{pyth\_exponent\_rep}}}{\text{teamRpG}^{\text{pyth\_exponent\_rep}} + \text{oppRpG\_rep}^{\text{pyth\_exponent\_rep}}}$$

$$\text{WAR\_rep} = (0.5 - \text{waa\_win\_perc\_rep}) \times \text{G}$$

$$\text{bWAR} = \text{WAA} + \text{WAA\_adj} + \text{WAR\_rep}$$

A.3.2 new_bWAR

$$\text{scaled\_avg\_runs\_above\_rep} = \left(\text{RpO\_replacement} - \frac{\text{teamRpG}}{\text{outs\_per\_game\_cnst}}\right)$$

$$\times (\text{GS} \times \text{outs\_per\_start\_rep} + \text{IPouts\_relief})$$

$$\text{new\_oppRpG\_rep} = \text{teamRpG} + \frac{\text{scaled\_avg\_runs\_above\_rep}}{\text{G}}$$

$$\text{new\_pyth\_exponent\_rep} = (\text{teamRpG} + \text{new\_oppRpG\_rep})^{0.285}$$

$$\text{new\_waa\_win\_perc\_rep} = \frac{\text{teamRpG}^{\text{new\_pyth\_exponent\_rep}}}{\text{teamRpG}^{\text{new\_pyth\_exponent\_rep}} + \text{new\_oppRpG\_rep}^{\text{new\_pyth\_exponent\_rep}}}$$

$$\text{new\_WAR\_rep} = (0.5 - \text{new\_waa\_win\_perc\_rep}) \times \text{G}$$

$$\text{scaled\_bp\_runs\_above\_avg} = \left(\frac{\text{teamRpG}}{\text{outs\_per\_game\_cnst}} - \text{bullpenRpO}\right)$$

$$\times \text{GS} \times (\text{outs\_per\_start} - \text{outs\_per\_start\_rep})$$

$$\text{oppRpG\_bp} = \text{teamRpG} - \frac{\text{scaled\_bp\_runs\_above\_avg}}{\text{G}}$$

$$\text{pyth\_exponent\_bp} = (\text{teamRpG} + \text{oppRpG\_bp})^{0.285}$$

$$\text{waa\_win\_perc\_bp} = \frac{\text{teamRpG}^{\text{pyth\_exponent\_bp}}}{\text{teamRpG}^{\text{pyth\_exponent\_bp}} + \text{oppRpG\_bp}^{\text{pyth\_exponent\_bp}}}$$

$$\text{WAA\_bp} = (\text{waa\_win\_perc\_bp} - 0.5) \times \text{G}$$

$$\text{new\_bWAR} = \text{WAA} + \text{WAA\_adj} + \text{new\_WAR\_rep} - \text{WAA\_bp}$$

A.4 Defining WAA_adj and the replacement level

A.4.1 WAA_adj

WAA_adj is a minor adjustment factor added in the final step of bWAR. It averages -0.092 and has max and min values of 0.016 and -0.191 in the observations. While no correct explicit formula for WAA_adj exists on Baseball Reference, multiple components are mentioned, including leverage, a constant averaging the yearly WAA to 0, and the presence of relief innings. As a result, I've been able to approximate WAA_adj with the following formula:

$$\text{WAA\_adj} \approx \text{WAA} \times \frac{\text{GR\_leverage\_index\_avg} - 1}{2} \times \frac{\text{IPouts\_relief}}{\text{IPouts}} + \text{IPouts} \times \text{WAA\_factor}$$

WAA_factor is the per-out constant centering WAA for the year at 0 and GR_leverage_index_avg is the pitcher's average leverage when entering a game. The resulting values differ from WAA_adj by an average of 0.035 and a maximum of 0.19, which can be improved upon slightly by tweaking WAA_factor.

For the purposes of this study, this approximation is close enough to official values to strongly suggest that WAA_adj is not affected by any of the proposed corrections, and thus can be left untouched in new_bWAR.

A.4.2 Replacement runs and outs_per_game_cnst

One of the bWAR steps that is not fully explained on Baseball Reference is how to calculate the difference in runs between replacement and average. Unlike WAA_adj, this value is crucial to the proposed correction and thus cannot be ignored or badly approximated. There are two variables in the .csv file that relate directly to this quantity: RpO_replacement and runs_above_rep. The former, a league-wide constant, is the number of runs per out the replacement allows. The latter is the number of runs a specific pitcher would save above the replacement.

For the purposes of calculating the original bWAR, runs_above_rep is all that is needed, since it becomes apparent that

$$\text{oppRpG\_rep} = \text{teamRpG} + \frac{\text{runs\_above\_rep} - \text{runs\_above\_avg}}{\text{G}}$$

and from there WAR_rep is found via PyPat. The quantity runs_above_rep − runs_above_avg, which I've named avg_runs_above_rep, represents the difference in runs allowed between the replacement and average, and is thus the quantity I want to calculate and modify in the correction[22]. The Baseball Reference pages give a few explanations as to how to find this value, but unfortunately, none of them contains a correct explicit formula for runs_above_rep, nor how that value is related to RpO_replacement.

To find this relation, I noticed that there are theoretically two different ways to calculate the difference in runs between average and replacement pitcher. The first is to take the difference between runs_above_rep and runs_above_avg. The second is to take the difference between RpO_replacement and teamRpG/(outs per game per team), since teamRpG is nominally the average performance by year and league, and then scaling the whole by IPouts. This allows the following equivalence:

$$\left(\text{RpO\_replacement} - \frac{\text{teamRpG}}{\text{outs\_per\_game\_per\_team}}\right) \times \text{IPouts} = \text{runs\_above\_rep} - \text{runs\_above\_avg}$$

Entering the average number of outs per game per team by league and year does not result in equivalent values on both sides, but gets close. For example, the average number of outs per game per team in AL '23 is approximately 26.64[23]. Using this value, the left side undershoots the right by an average of -0.26 runs. This is close but still results in an extra run or two being artificially attributed to the replacement, depending on the pitcher.

Fortunately, outs_per_game_per_team (or rather, whatever that value is supposed to represent, which I'm calling outs_per_game_cnst) is the only variable in the equation whose exact value is not given in the .csv file. This allows us to solve for the constant:

$$\text{outs\_per\_game\_cnst} = \frac{\text{teamRpG}}{\text{RpO\_replacement} - \frac{\text{runs\_above\_rep} - \text{runs\_above\_avg}}{\text{IPouts}}}$$

[22] Theoretically, runs_above_rep − runs_above_avg can be plugged into the correction directly without bothering with outs_per_game_cnst. The problem is that this doesn't explain the relationship between runs_above_rep and RpO_replacement, or even if such a relationship exists. Since RpO_replacement is the official replacement level, it felt necessary to create a correction that took RpO_replacement into account, or at least to accurately describe the relationship between it and runs_above_rep.

[23] This value can be found by dividing the number of outs pitched by 2 * 1,215, the number of games. Values taken from https://www.baseball-reference.com/leagues/AL/2023.shtml

The only issue with this formula is that it's based on the individual player's IPouts stat. For players with low IPouts counts, this results in an outs_per_game_cnst value that differs noticeably from that of other players. Fortunately, as IPouts increases, the value converges, presumably towards the "true" constant. Thus, we need only run this calculation for the player with the highest IPouts count. This approach is illustrated in Figure 7; data is from NL '24. The values are the outs_per_game_cnst calculated individually for each player, while the red line represents the outs_per_game_cnst value for the player with the highest IPouts count. This is the value I use in the official formula and correction.

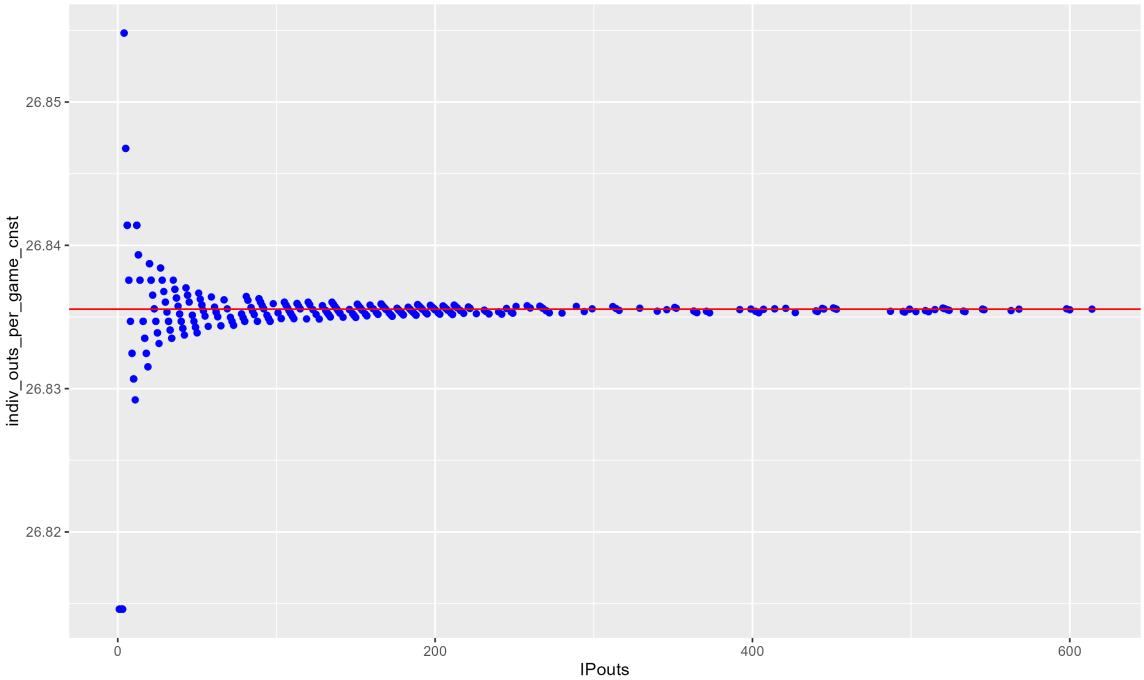


Figure 7: Individual outs_per_game_cnst values for NL '24

The average difference between the left and right side when using outs_per_game_cnst instead of the actual number of outs per game per team is now 0.000026 across the entire data set, which disappears once rounding takes place. This allows us to state with certainty that

$$\begin{aligned}\text{avg\_runs\_above\_rep} &= \text{runs\_above\_rep} - \text{runs\_above\_avg} \\ &= \left(\text{RpO\_replacement} - \frac{\text{teamRpG}}{\text{outs\_per\_game\_cnst}}\right) \times \text{IPouts}\end{aligned}$$

and to use outs_per_game_cnst in both the official formula and the correction directly.

I make two final observations. First, this result proves that the average/replacement difference, unlike the pitcher/average difference, does not take into account the pitcher's circumstances. In fact, the only pitcher-specific metric it uses is the number of outs pitched. This creates two separate average baselines: the replacement is compared to an average pitcher on an average team, but the starter is compared to an average pitcher in circumstances specific to the starter. These two baselines may vary considerably depending on the player but are treated identically by bWAR.

Second, outs_per_game_cnst is consistently a few decimal points higher than the corresponding outs per game per team. This recalls the runs-to-win page, which states that when computing the outs per game for pitchers, "we compute outs recorded per game for the season (capped at 26.8) and then pad the remainder of the game with league average run prevention"[24]. I am not sure what the "league

[24]Sports Reference LLC, "Baseball-Reference.com WAR Explained, Converting Runs to Wins"

average run prevention” refers to; the term is only mentioned once on the page and I have not found it discussed elsewhere on Baseball Reference. Additionally, this explanation is in the section on bWAR 2.0, an outdated version of the formula. Nonetheless, I believe this is how outs_per_game_cnst is found organically: take the average number of outs per start per team and add a small league-wide constant based on some defensive or pitching metric. I haven’t been able to confirm this theory.

A.5 Constants by year and league

<table>
<tr><th>Year</th><th>League</th><th>RpO_replacement</th><th>teamRpG</th><th>outs_per_game_cnst</th><th>bullpenRpO</th><th>outs_per_start_rep</th></tr>
<tr><td rowspan="2">2022</td><td>AL</td><td>0.188</td><td>4.25</td><td>26.83</td><td>0.155</td><td rowspan="6">14.87</td></tr>
<tr><td>NL</td><td>0.194</td><td>4.40</td><td>26.89</td><td>0.166</td></tr>
<tr><td rowspan="2">2023</td><td>AL</td><td>0.203</td><td>4.61</td><td>26.95</td><td>0.172</td></tr>
<tr><td>NL</td><td>0.208</td><td>4.71</td><td>26.85</td><td>0.170</td></tr>
<tr><td rowspan="2">2024</td><td>AL</td><td>0.193</td><td>4.38</td><td>26.98</td><td>0.165</td></tr>
<tr><td>NL</td><td>0.198</td><td>4.48</td><td>26.84</td><td>0.164</td></tr>
</table>

Table 5: Constant table