Simulation: Win Probability in Simplified Football

* But how reliable are Football WP estimates given from an XGBoost or random forest model? Just because they say $\widehat{WP}_{Go} = .63$, $\widehat{WP}_{FG} = .61$, $\widehat{WP}_{punt} = .59$ doesn't mean these values are <u>Right</u> or are any good.

* Need Uncertainty Quantification — prediction interval $\hat{I}(x) = [\widehat{WP}_L(x), \widehat{WP}_u(x)]$ so that most of the time $WP(x) \in \hat{I}(x)$

* If prediction intervals are wide

e.g. $\widehat{WP}(x) = .63 \in [.5, .76]$

then we're not confident that our estimate is good

If not, e.g. $\widehat{WP}(x) = .63 \in [.62, .64]$

then we are confident

* How to get prediction intervals for blackbox machine learning models like XGBoost or Random Forest?

→ A fundamental open problem in machine learning today...

* Our approach: **Bootstrap Uncertainty Quantification**

```
Input: training set $T$
Hyperparameters: $B$ = # bootstrapped datasets
                 $m$ = # rows in each bootstrapped dataset
for $b = 1, \ldots, B$:
    $T^{(b)}$ = resample $m$ rows $(x_i, y_i)$
                from $T$ with replacement
    fit $\hat{f}^{(b)} = \hat{f}(T^{(b)})$
Output $\hat{I}(x) = [$2.5th quantile of $\hat{f}^{(b)}(x)$, 97.5th quantile of $\hat{f}^{(b)}(x)$]
```

* Intuitively, bootstrapped CI works because $T^{(b)} \overset{\text{resampled}}{\sim} T$ approx. has the same distribution as $T \sim$ underlying true population dist.

* Problem: the original bootstrap (sampling Rows with replacement) assumes i.i.d. data.

Is our data i.i.d.? No!!!

Recall our dataset:

i = index of i^{th} play in dataset of NFL plays

$y_i = 1$ if team with possession on play i wins, else 0

y_i is highly autocorrelated: for each play i in game $g(i)$, $y_i \equiv y_{g(i)}$, i.e. they share the same value of the response column.

There is one independent draw per game!

Dataset: 500,000 plays
4,000 independent draws

- * Instead, use a Cluster Bootstrap: resample clusters (games) with replacement, not rows (plays). Because that is how the data is generated!

- * So, we'll use a cluster bootstrap to obtain CI. How do we set the hyperparameters B and m? How do we know if our CI are actually good (i.e., good coverage)?

* Simulation:

1. Create a fake simplifed version of football in which true WP is known

2. Generate a fake observational dataset of football plays

3. Fit $\widehat{WP}$ using machine learning on our synthetic historical dataset, and create bootstrapped CI

4. Check whether our CI have satisfactory coverage (and whether $\widehat{WP}$ is unbiased).

3.2 Problems with existing win probability models

Football analysts see a dataset of $511,264$ plays from 2006 to 2021 and think this is enough data to fit accurate WP models. This, however, is not true because football data is highly autocorrelated: *every game has only one winner*. Formally, the binary response variable y_i of the i^{th} play indicates whether the team with possession won the game.[12] Crucially, the reponse values are not independent, as all plays from the same game share the *same draw* of the response column. On this view, the effective sample size is much closer to $4,101$, the number of non-tied games from 2006 to 2021. This is nowhere near enough data to experience the full variability of the nonlinear and interacting variables of score differential, time remaining, point spread, yardline, yards to go, timeouts, etc. In fitting win probability models, we are in a limited-data context, and as such we expect wide confidence intervals and some bias in win probability point estimates.

3.3 Simulation study

To better understand how autocorrelation affects estimating win probability from observational data, we conduct a simulation study. Specifically, we create a simplified version of football in which the true win probability at each game-state is known. Then, we see how well existing methods recover the true win probability. In particular, we measure the average error between true and estimated WP, and we compare the coverage and lengths of various bootstrapped WP confidence intervals. In our simulations, we find that XGBoost fit from autocorrelated observational data recovers the general trend of WP, with a mean absolute error of less than 2% WP. The standard bootstrap, which ignores the autocorrelated nature of the dataset, yields confidence intervals which are too narrow, resulting in subpar coverage. On the other hand, the randomized cluster bootstrap, which accounts for autocorrelation, obtains approximately 90% frequentist coverage of the true win probability via subtantially wide confidence intervals with an average width of 8% WP. Our primary takeaway is that win probability models fit from an autocorrelated historical dataset of $4,101$ games are subject to substantial uncertainty.

Rules of the game. Our game, a simplified form of football, begins at midfield. Each play, the ball moves left or right by one yardline with equal probability. If the ball reaches the left (right) end of the field, team one (two) scores a touchdown, worth one point. The ball resets to midfield after each touchdown. After N plays, the game ends. If the game is still tied after N plays, a fair coin is flipped to determine the winner. We discuss the formal mathematical specification of the game in

[12] The response variable for fitting EP models from observational data is also autocorrelated, as plays are clustered into *epochs* (plays which share the same next score outcome). Nevertheless, our dataset of football plays from 2006 to 2021 contains $47,874$ epochs, and each epoch contains an average of about 11 plays. We use the same methods developed later in this section (e.g., the randomized cluster bootstrap) to quantify uncertainty in EP estimates, and we find that autocorrelation impacts EP models to a significantly smaller degree than it affects WP models.

Appendix S2.1. Additionally, we explicitly compute true win probability as a function of timestep, field position, and score differential using dynamic programming (see Appendix S2.1 for details).

Simulation methodology. 25 times, we simulate G games, each with N plays per game. We use $L = 4$ yardlines so that the average number of plays between each score is similar to that of a real football game. This yields 25 simulated datasets of simplified football plays, each of the form

$$\mathscr{D} = \{(n, X_{gn}, S_{gn}, y_{gn}) : n = 1, ..., N \text{ and } g = 1, ..., G\}. \tag{3.4}$$

For each play of game g, we record the timestep n, the field position X_{gn}, the score differential S_{gn}, and a binary variable y_{gn} indicating whether the team with possession wins the game. The response variable y is autocorrelated, as each play within the same game shares the same random draw of y.

On each simulated dataset, we use machine learning to estimate win probability as a function of timestep n, field position x, and score differential s,

$$\widehat{\mathsf{WP}}(n,x,s) = \mathsf{XGBoost}(\mathscr{D})(y|n,x,s). \tag{3.5}$$

We then compute the mean absolute error between the true and estimated win probabilities averaged over the 25 simulations. We also compare the coverage and lengths of the WP confidence intervals produced by various bootstraps, discussed below, averaged over the 25 simulations.

Bootstrap methodology. We compare the coverage and lengths of the WP confidence intervals produced by the standard bootstrap, cluster bootstrap, and randomized cluster bootstrap, averaged over the 25 simulations. In the standard bootstrap, which assumes each row (play) of the dataset is independently drawn., each of B bootstrapped datasets are formed by resampling N plays with replacement. In the cluster bootstrap, each of B bootstrapped datasets are formed by resampling G' games with replacement, keeping each observed row within each resampled game. Finally, in the randomized cluster bootstrap, each of B bootstrapped datasets are formed by resampling G' games with replacement, and within each game resampling plays with replacement. To acheive better coverage, we resample half as many games as in the original dataset, $G' = G/2$. Then, for each bootstrap method, we fit a WP model WP_b to each bootstrapped dataset b. The confidence interval for the WP estimate at game-state $\mathbf{x}$ is defined by the 2.5^{th} and 97.5^{th} quantiles of $\{\mathsf{WP}_1(\mathbf{x}), ..., \mathsf{WP}_B(\mathbf{x})\}$.

Simulation results. We report the results of our simulation study in Table 3. The first row reports results for which our simulated datasets consist of $G = 4101$ games and $N = 53$ plays per game, which matches the number of games and the average number of first down plays in our dataset of real football plays. Each game in these datasets consists of $K = 53$ autocorrelated plays per game.

The second row reports results for which our simulated datasets consist of $G = 4101 \cdot 53$ games and $N = 53$ plays per game, but we keep just $K = 1$ play per game. In other words, those datasets consist of 217353 plays with an i.i.d. response column.

G	N	K	MAE bt WP and $\widehat{\text{WP}}$	CI covg. SB	CI covg. CB	CI covg. RCB	CI length SB	CI length CB	CI length RCB
4101	53	53	0.0179	0.73	0.85	**0.90**	0.048	0.067	**0.079**
$4101 \cdot 53$	53	1	0.0164	0.78	0.78	0.78	0.049	0.049	0.049

Table 3: Simulation study results. SB means standard bootstrap, CB means cluster bootstrap, and RCB means randomized cluster bootstrap.

In the simulation study with autocorrelation ($K = 53$), the mean absolute error (MAE) between the true WP and the WP estimated by XGBoost is less than 2% over average.[13] So, XGBoost recovers the general trend of the true WP. In the simulation study without autocorrelation ($K = 1$), the MAE is similar but slightly smaller. This suggests that most of the bias induced by fitting WP from observational data is the result of having limited data, not from the autocorrelation.

The length and coverage of win probability confidence intervals, on the other hand, are significantly impacted by autocorrelation. In the simulation study with autocorrelation ($K = 53$), the standard bootstrap, which ignores autocorrelation, produces confidence intervals which are too narrow at an average width of about 5% WP, leading to a subpar 73% coverage. The cluster bootstrap produces wider confidence intervals at an average width of about 7% WP, leading to a higher 85% coverage. The randomized cluster bootstrap produces even wider confidence intervals at an average width of about 8% WP, leading to a satisfactory frequentist coverage of 90% over average. Additionally, coverage from the randomized cluster bootstrap is similar across all values of true win probability except near 0 and 1.[14] To increase coverage at the extremes, we widen our confidence intervals when $\widehat{\text{WP}} < 0.025$ to have a lower bound of 0 and when $\widehat{\text{WP}} > 0.975$ to have an upper bound of 1. Also, average confidence interval length from the randomized cluster bootstrap is at most 12% for some values of true WP, and C.I. length decreases as true WP moves towards the extremes.[15]

In the simulation study without autocorrelation ($K = 1$), on the other hand, each bootstrap method is identical and yields an average confidence interval length of about 5% WP (similar to the average C.I. length from the standard bootstrap on autocorrelated data). The average frequentist coverage is 78%; to increase coverage we could widen the confidence intervals by resampling fewer than $(G/2) \cdot N$ plays in the standard bootstrap.

[13] The MAE is smaller than about 3.5% WP across all values of true win probability. See Figure S16a of Appendix S2.1 for details.

[14] See Figure S16b of Appendix S2.1 for details.

[15] See Figure S16c of Appendix S2.1 for details.

S2 Win probability details

S2.1 Simulation study details

Generating plays. Formally, the outcome of the n^{th} play of the g^{th} game is

$$\xi_{gn} \overset{iid}{\sim} \pm 1. \tag{S1}$$

The game starts at midfield, $X_{g0} = L/2$, and the game begins tied, $S_{g0} = 0$. The field position at the start of play n is

$$X_{g,n+1} := \begin{cases} X_{gn} + \xi_{gn} & \text{if } 0 < X_{gn} + \xi_{gn} < L \text{ (not a TD)} \\ L/2 & \text{else,} \end{cases} \tag{S2}$$

and the score differential at the start of play n is

$$S_{g,n+1} := \begin{cases} S_{gn} + 1 & \text{if } X_{gn} + \xi_{gn} = 0 \text{ (TD)} \\ S_{gn} - 1 & \text{if } X_{gn} + \xi_{gn} = L \text{ (opp. TD)} \\ S_{gn} & \text{else.} \end{cases} \tag{S3}$$

The response column *win* is

$$y_{gn} \equiv y_{g,N+1} := \begin{cases} 1 & \text{if } S_{g,N+1} > 0 \\ 0 & \text{if } S_{g,N+1} < 0 \\ \text{Bernoulli}(1/2) & \text{else (overtime).} \end{cases} \tag{S4}$$

As in our dataset of real football plays, this response column is highly autocorrelated – plays from the same game share the same draw of the winner of the game.

Generating observational data. We create a dataset of plays from G games. Each game consists of N plays, and the field consists of L yardlines. The results from each game yield a simulated dataset

$$\mathscr{D} = \{(n, X_{gn}, S_{gn}, y_{gn}) : n = 1, ..., N \text{ and } g = 1, ..., G\}. \tag{S5}$$

True win probability. The true win probability

$$\mathsf{WP}(n, x, s) := \mathbb{P}(S_{g,N+1} > 0 | X_{gn} = x, S_{gn} = s) \tag{S6}$$

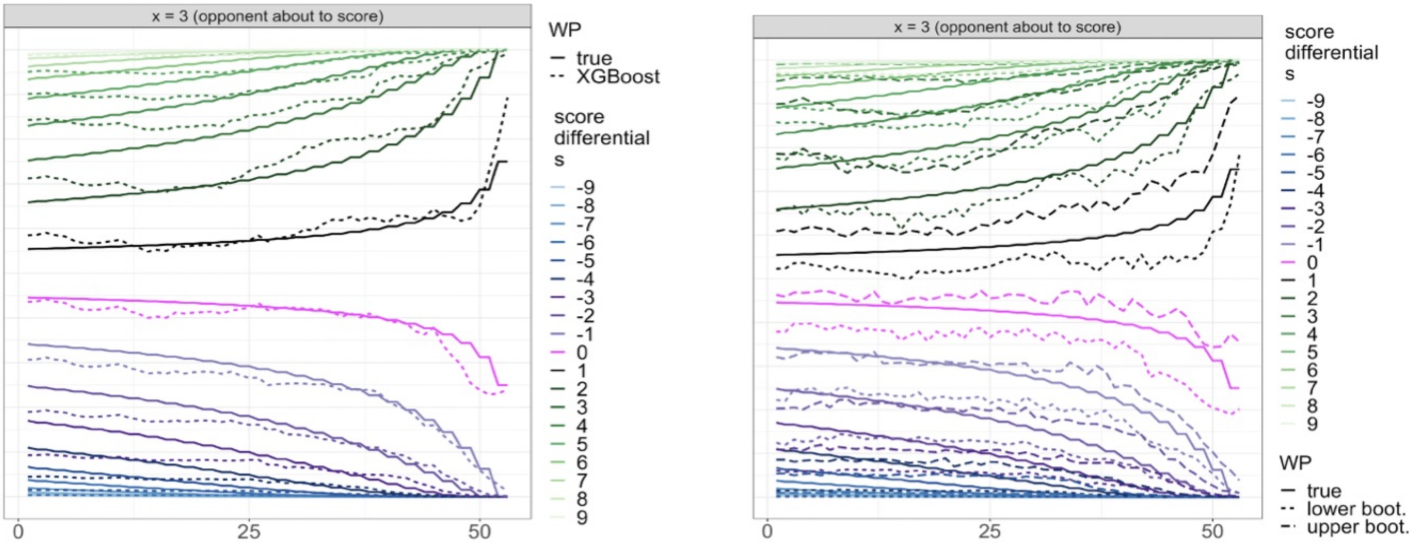
of our simplified version of football is computed explicitly using dynamic programming,

$$\mathsf{WP}(N+1,x,s) = \begin{cases} 1 & \text{if } s>0 \\ 1/2 & \text{if } s=0 \\ 0 & \text{if } s<0, \end{cases} \tag{S7}$$

and

$$\mathsf{WP}(n-1,x,s) = \begin{cases} \frac{1}{2}\mathsf{WP}(n,\frac{L}{2},s+1)+\frac{1}{2}\mathsf{WP}(n,x+1,s) & \text{if } x=1 \\ \frac{1}{2}\mathsf{WP}(n,x-1,s)+\frac{1}{2}\mathsf{WP}(n,\frac{L}{2},s-1) & \text{if } x=L-1 \\ \frac{1}{2}\mathsf{WP}(n,x-1,s)+\frac{1}{2}\mathsf{WP}(n,x+1,s) & \text{else.} \end{cases} \tag{S8}$$

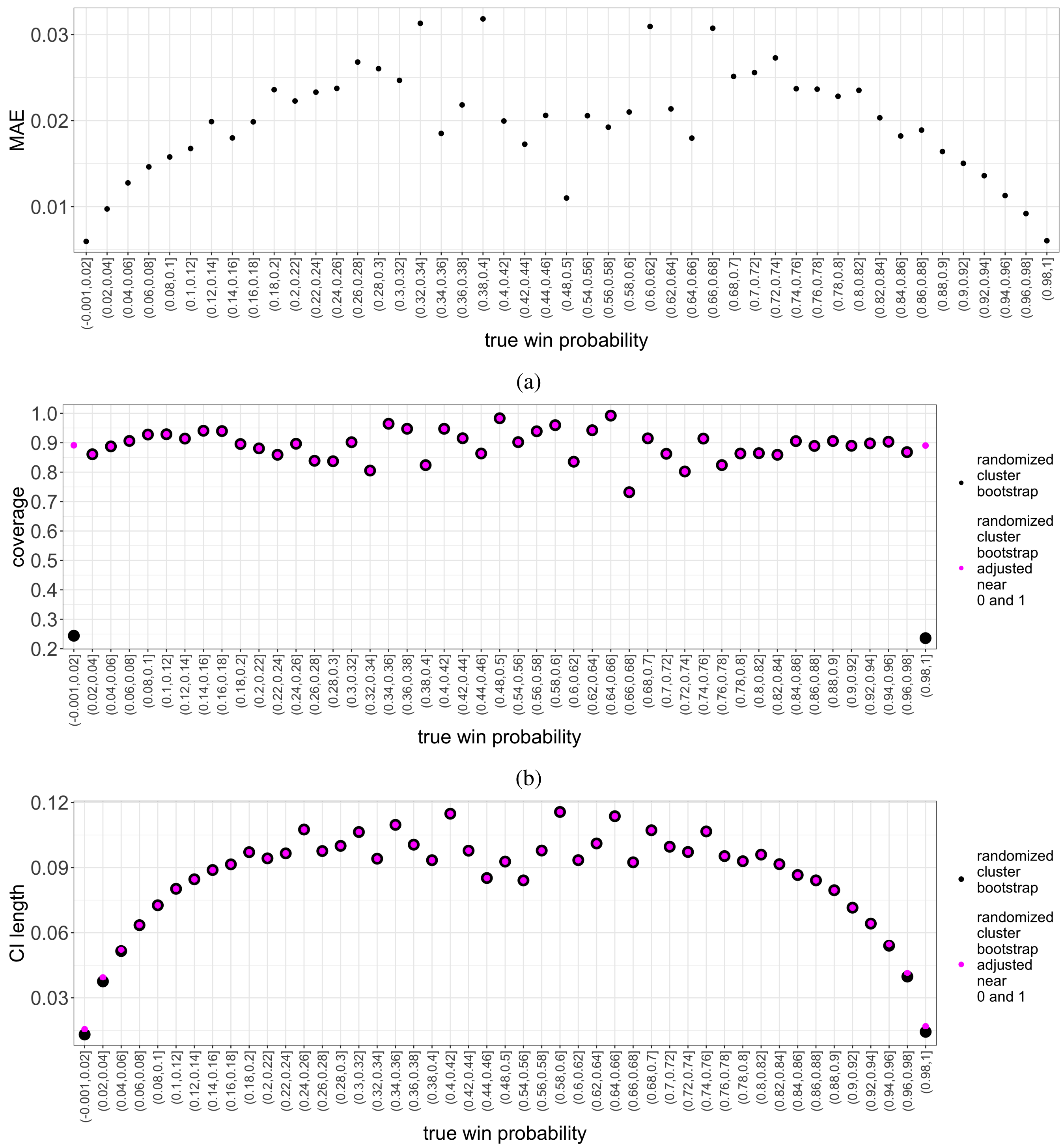
Visualizing the simulation study results. In Figure S16 we visualize the MAE of WP estimates and the confidence interval lengths and coverages, averaged over all of the simulations. In Figure S17 we visualize the WP point estimates and bootstrap confidence intervals for one simulation.



(a)

(b)

Figure S17: On the left, we visualize the error between estimated WP (dotted line) and true WP (solid line). On the right, we visualize the WP confidence intervals (dotted line) produced by the randomized cluster bootstrap and the true WP (solid line). Both figures display the results from one simulation and at yardline $x = 3$.



(a)

(b)

(c)

Figure S16: As a function of true WP, MAE of true and estimated WP (Figure (a)), coverage of true WP by randomized cluster bootstrap (Figure (b)), and confidence interval length of randomized cluster bootstrap (Figure (c)).