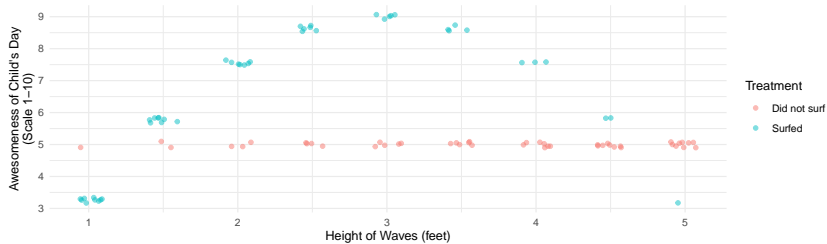


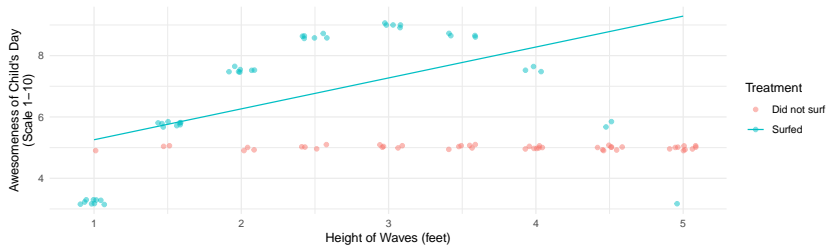
## Doubly Robust Slides

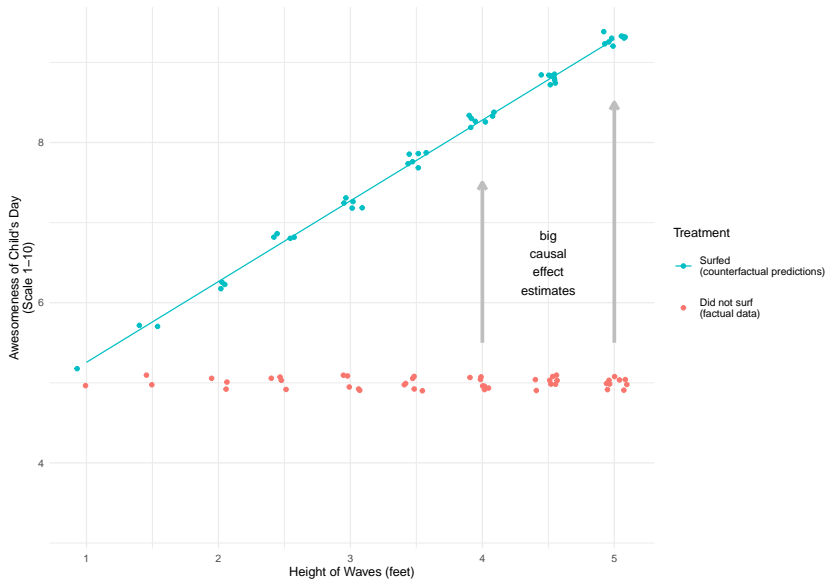


# A motivating example



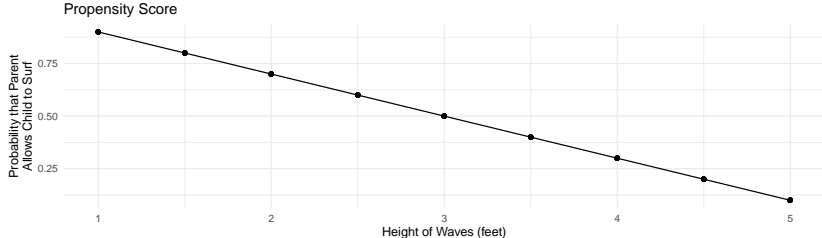
$$\tau = \frac{1}{n_{\text{Not Surfed}}} \sum_{i:A_i=\text{Not Surfed}} (Y_i^{\text{Surfed}} - Y_i^{\text{Not Surfed}})$$





We are always the child

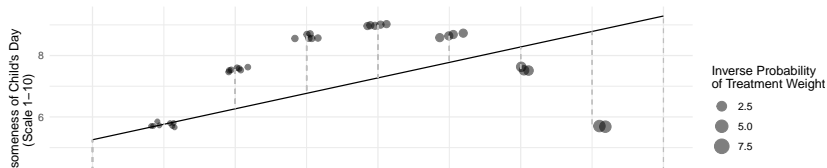
- ▶ model is too simple for the world
- ▶ model is always wrong



To be even more concrete, consider 5-foot wave days. The child knows that they surfed only 10% of these days. Every day surfed really stands in for 9 days not surfed. If we were estimating the average treatment effect on the untreated by weighting, we would weight the 5-foot day by

$$P(\text{Not Surfed} \mid X = 5) / P(\text{Surfed} \mid X = 5) = .9 / .1 = 9.$$

We therefore might estimate the weighted error of the linear model, weighted by these weights.







## Targeted learning

AIPW is not the only way to update a model. Another option is called targeted learning (Van der Laan and Rose 2011). We first introduce targeted learning through one concrete example, then generalize the procedure in the sections that follows.

### Modeling $Y^1$ for the ATC

In the surfing example, our goal is to estimate the mean outcome under surfing, for the observations on days when there was surfing. We begin with the child's initial fit: linear regression. Following notation that is common in targeted learning,<sup>1</sup> we will refer to this regression line as  $\hat{Q}^0$ ,

$$\underbrace{\hat{Q}^0(\vec{x})}_{\text{The 0 superscript indicates an untargeted initial estimate}} = \hat{E}(Y \mid A = 1, \vec{X}) = \hat{\alpha} + \hat{\beta}\vec{x}$$

The 0 superscript  
indicates an untargeted  
initial estimate

where  $\hat{\alpha}$  and  $\hat{\beta}$  are the OLS coefficients when modeling  $Y$  given  $X$  among the treated observations.



## Sample splitting for machine learning

Under classical statistical approaches to inference, one often worries about model misspecification. The problem of model misspecification is that if one approximates  $E(Y \mid A, \vec{X})$  by an additive regression and the true response function is nonlinear, then the model will be an inconsistent estimator of  $E(Y \mid \vec{X} = \vec{x})$  for at least some  $\vec{x}$ . Double robustness solves this problem: as long as  $E(Y \mid \vec{X})$  or  $P(A = 1 \mid \vec{X})$  is estimated by a consistent estimator, then the causal effect estimate is consistent.

Machine learning approaches seem to upend this logic: flexible models can be consistent estimators by construction. Without any assumption of a statistical model, a random forest can yield a consistent estimator of  $E(Y \mid A, \vec{X})$  and  $P(A = 1 \mid \vec{X})$ . By consistent, we mean that the forest would come to equal these estimands in an infinite sample. Does double robustness then have any use?

The problem with flexible machine learning estimators is that they converge to the true response surface at slower rates than parametric regression models. In finite samples, the estimators will

# Exercises

Carry out the child's analysis by

- ▶ an outcome model.
- ▶ inverse probability weighting, with weights estimated by logistic regression
- ▶ augmented inverse probability weighting