Problem Set 1

Mitchell Linegar

2020-04-29

Contents

Part One

This problem set examines the welfare data set. Throughout, the treatment variable will be referred to as W and the outcome variable will be referred to as Y.
## Collaborators I worked closely on this problem set with Ayush Kanodia. I also worked with Kaleb Javier Pena and Haviland Sheldahl-Thomason, and indicate where we collaborated on code.

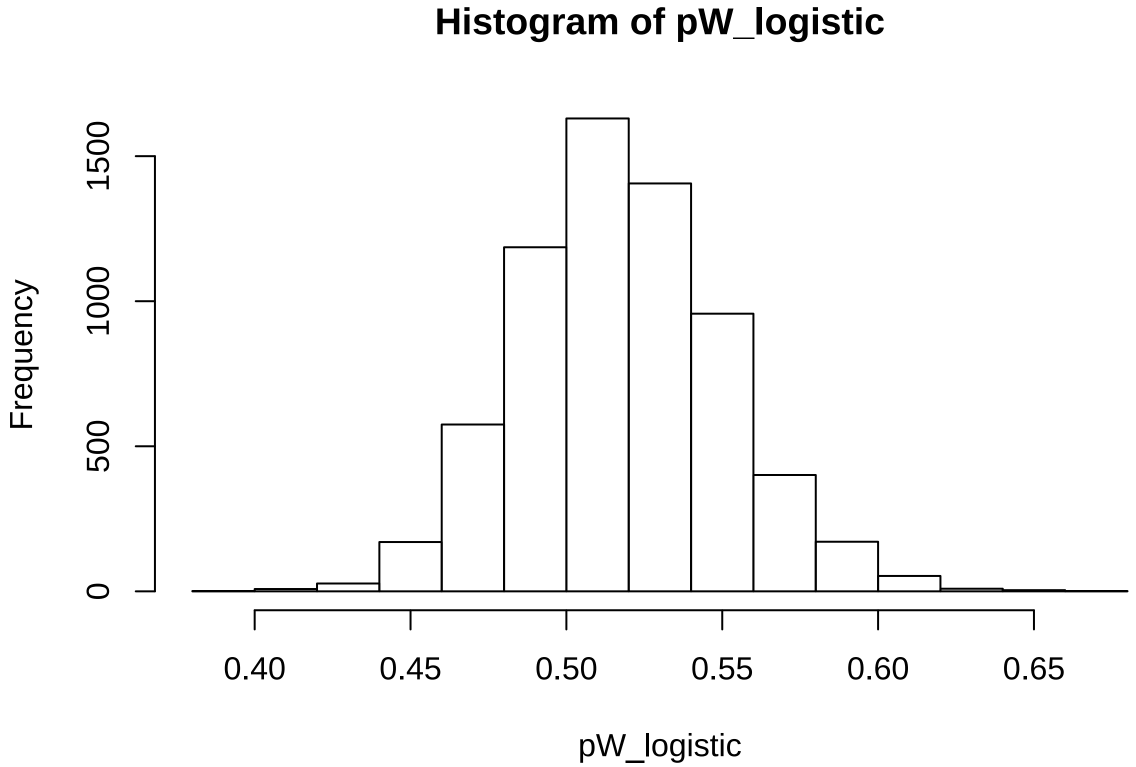
Pre-Processing the Data

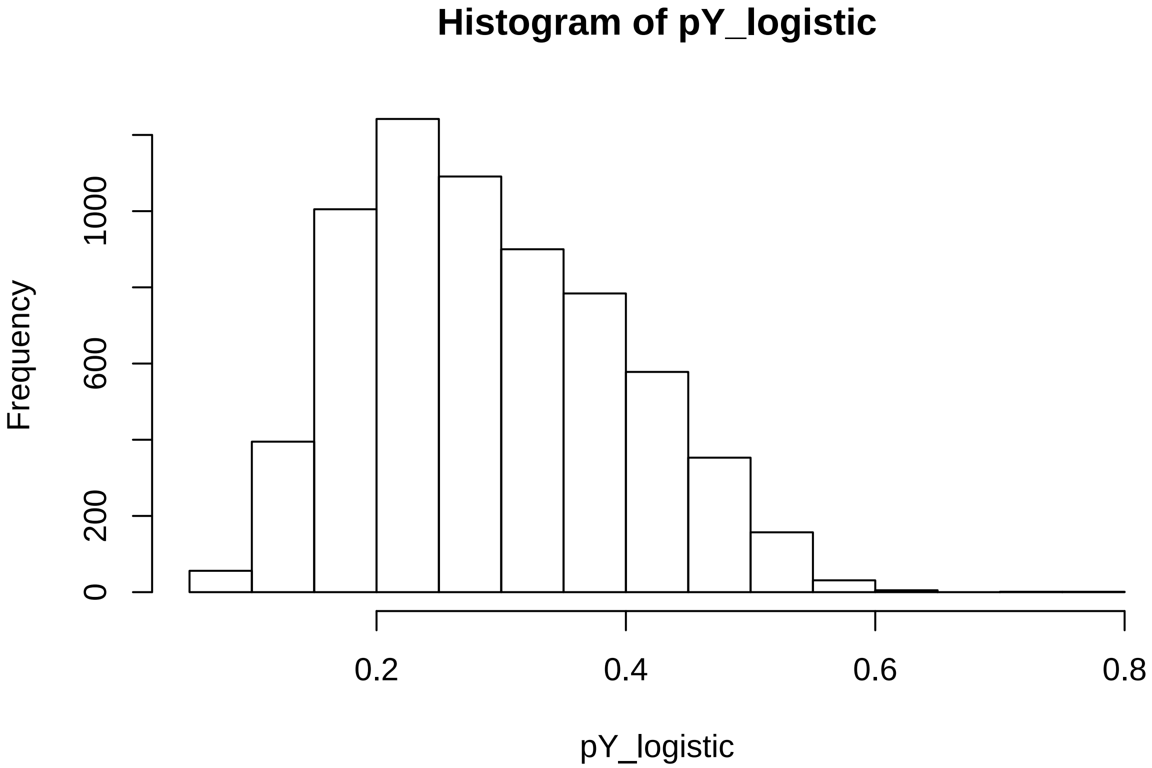
I use code from a set of AtheyLab tutorials, which include the following note:
> The datasets in our github webpage have been prepared for analysis so they will not require a lot of cleaning and manipulation, but let's do some minimal housekeeping. First, we will drop the columns that aren't outcomes, treatments or (pre-treatment) covariates, since we won't be using those. Specifically, we keep only a subset of predictors and drop observations with missing information.

Testing Assumptions

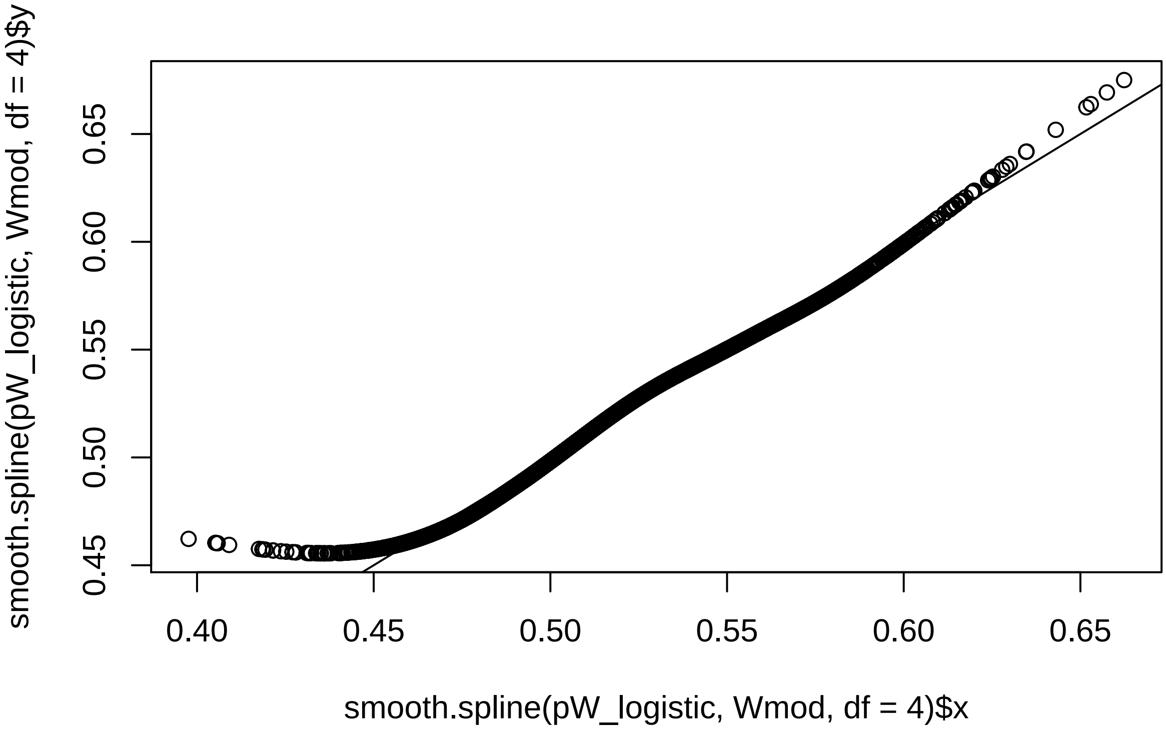
Here we test some of our traditional causal inference assumptions.

As a first step, we plot logistic predictions of the probabilities our treatment pW and our outcome pY. We see that treatment assignment appears to follow a normal distribution, and that our outcome has an average unconditional probability of 0.29.





We now produce a plot comparing predicted and actual treatment assignment. This plot is provided mostly for comparison (this is the plot the tutorial has); future plots of this nature will be done with `ggplot` to make their options more explicit.



RCT Analysis

We now report the (presumably true) treatment effect $\hat{\tau}$ from the randomized experiment:

```
##      ATE lower_ci upper_ci
##   -0.381   -0.401   -0.361
```

Introducing Bias

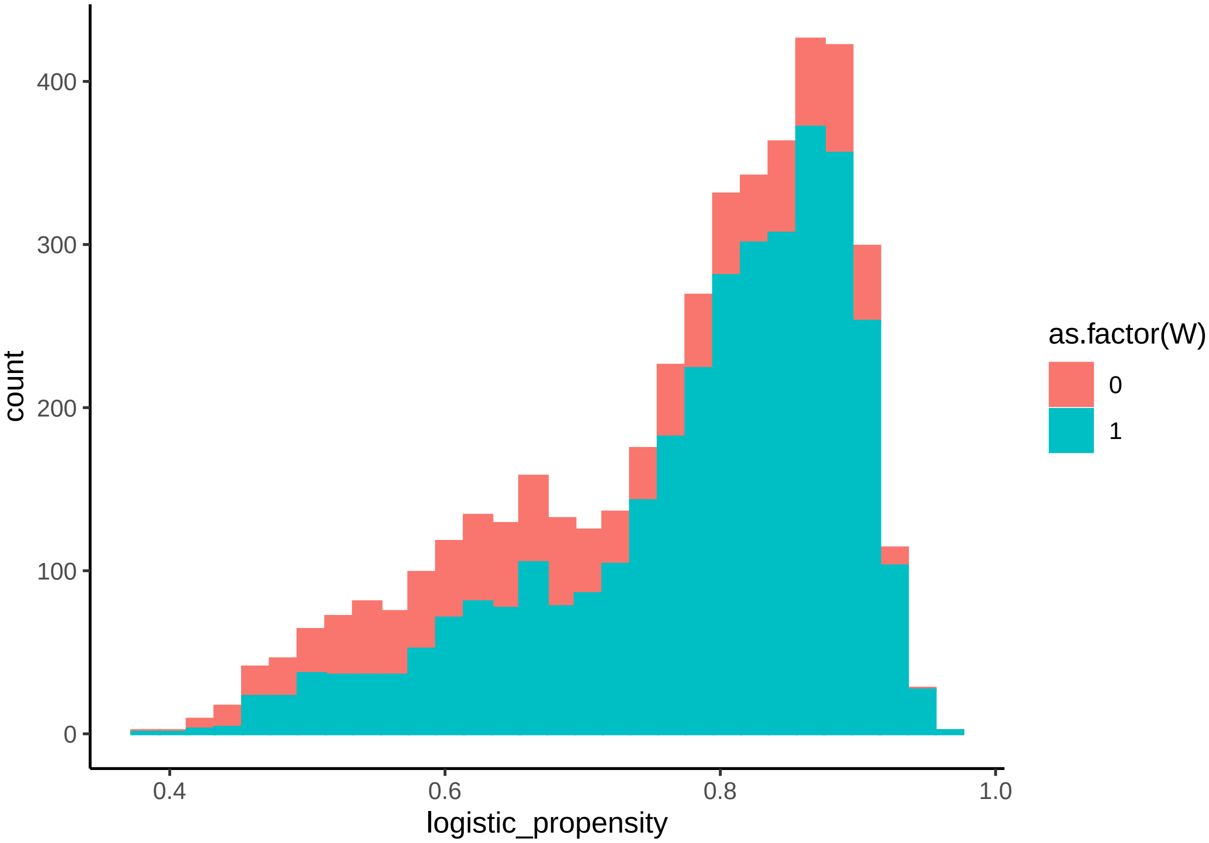
We now introduce sampling bias in order to simulate the situation we would be in if our data was from an observational study rather than a randomized experiment. This situation might arise due to sampling error or selection bias, and we will be able to see how various methods correct for this induced bias. To do, so, we under-sample treated units matching the following rule, and under-sample control units in its complement: - Independents on closer to the Democratic party (`partyid < 4`) - Who have at least a college degree (`educ >= 16`)

We remove 40 percent of observations in these sets.

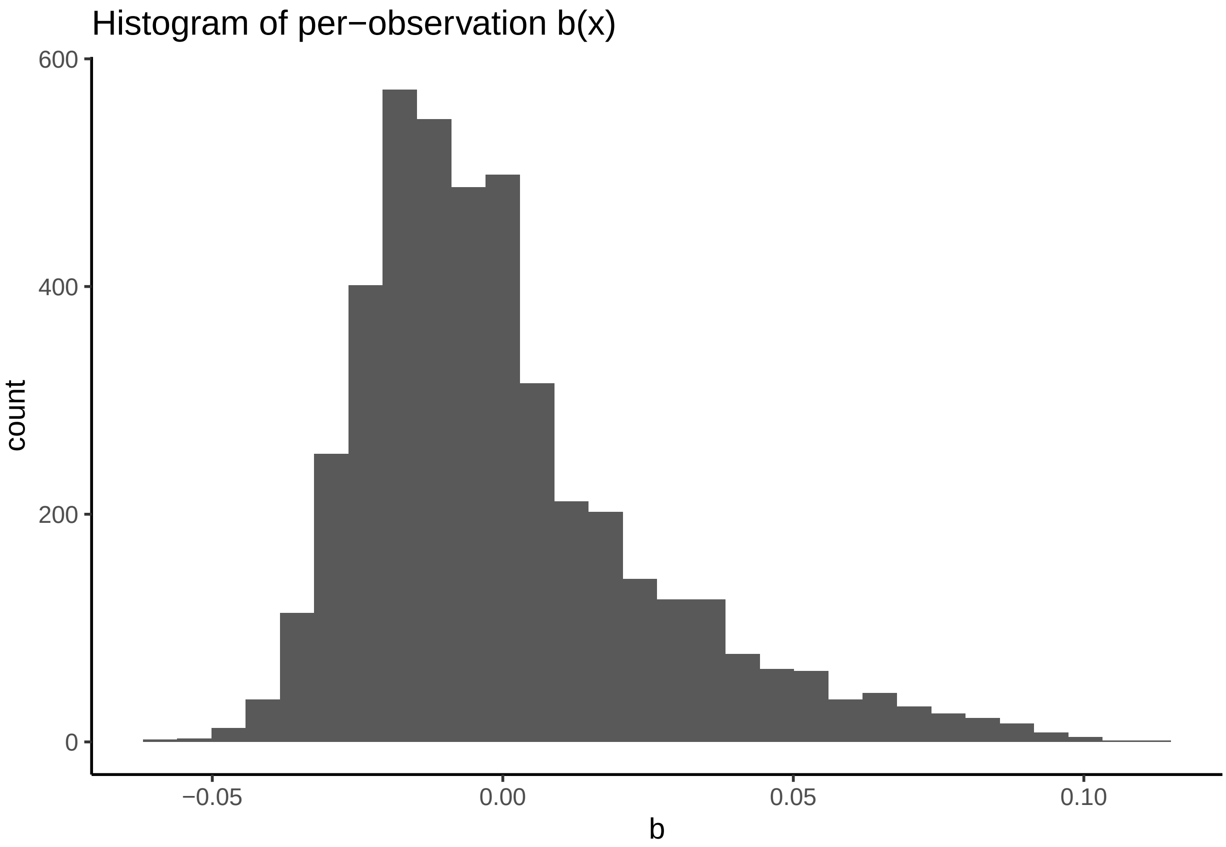
The difference in means is now biased, and significantly outside the confidence interval indicated by the RCT. Check if difference in treatment effect estimates is substantial

```
##      ATE lower_ci upper_ci
##   -0.291   -0.311   -0.271
```

We now plot (logistic) propensity scores, showing that we still have overlap after removing observations. We may be somewhat concerned about the small number of observations with propensities close to one; however, they are small in number and not exactly one so we are leaving them in for now.



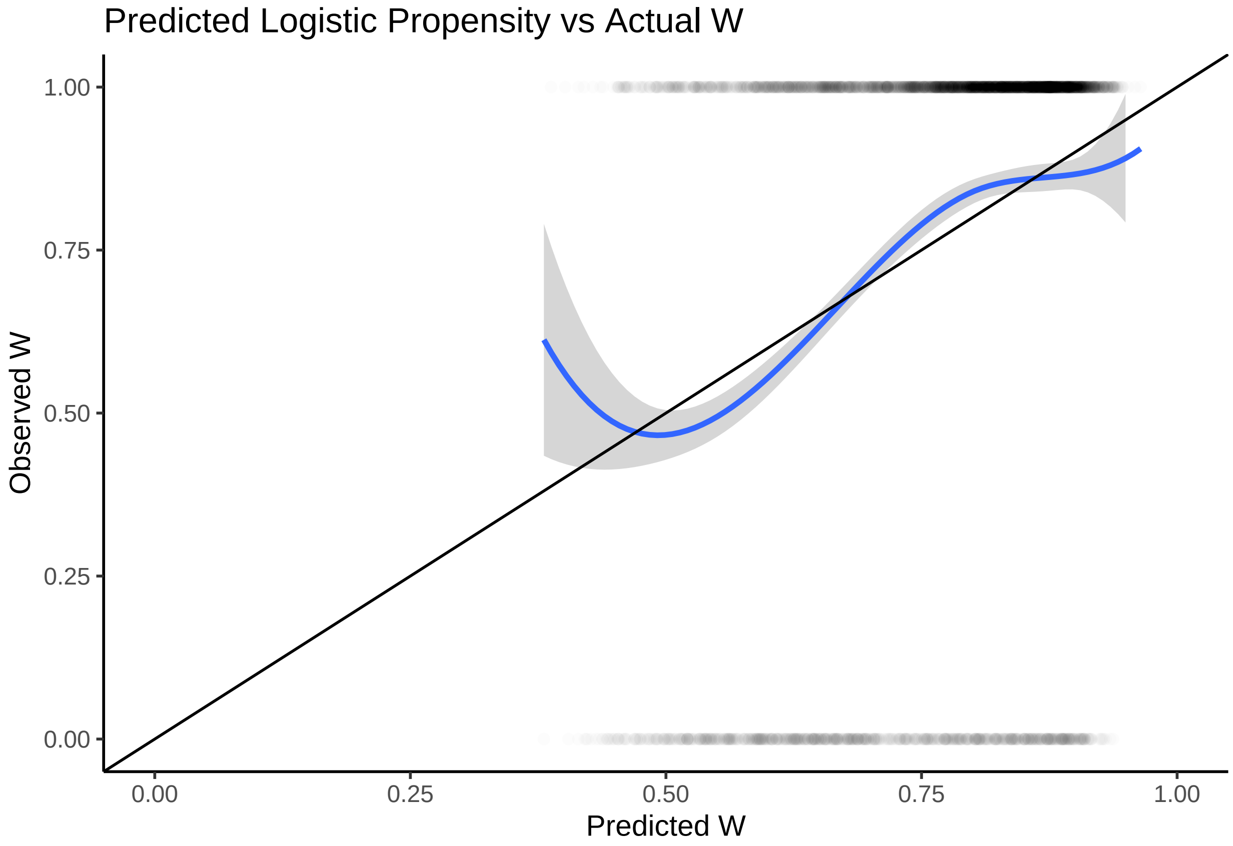
Next we plot the bias function $b(X)$ following Athey, Imbens, Pham and Wager (AER P&P, 2017, Section IIID). We plot $b(x)$ for all units in the sample, and see that the bias seems evenly distributed around zero. We see that bias for most observations is close to zero.

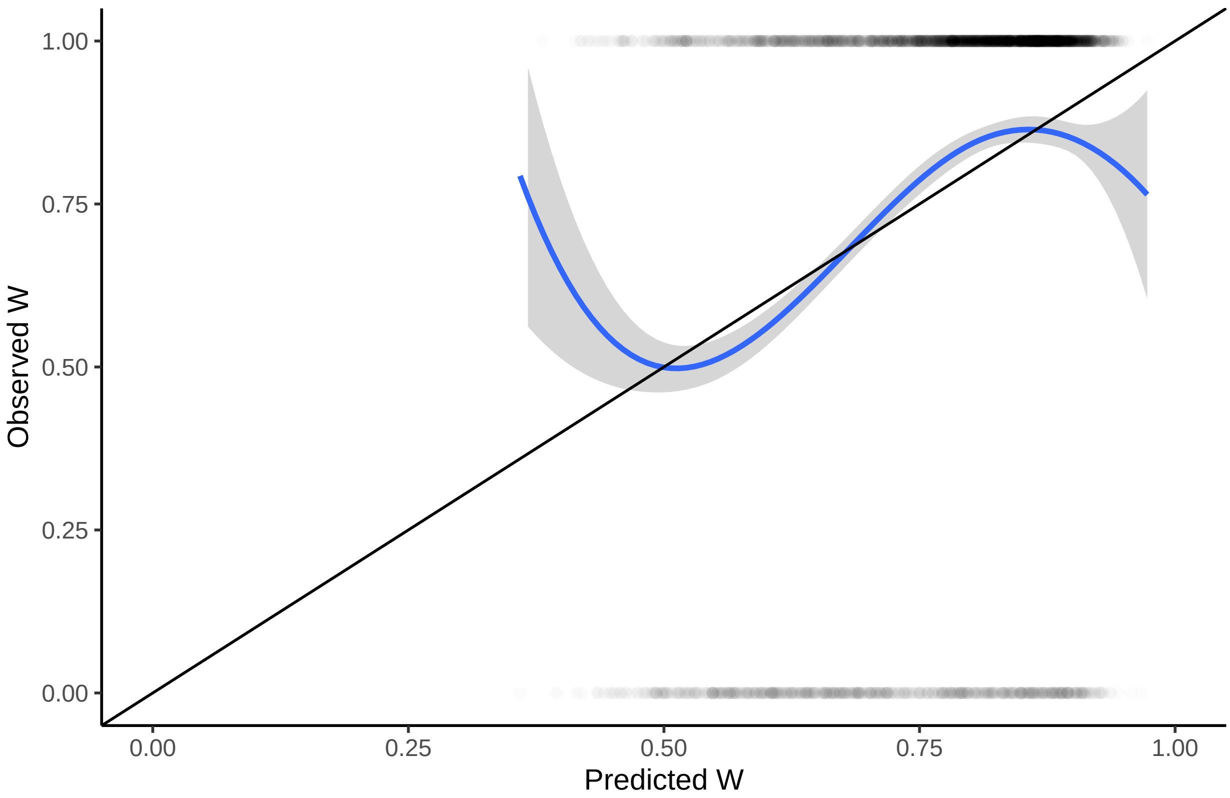

Histogram of per−observation b(x)
count
600
400
200
0
−0.05
0.00
0.05
0.10
b

Estimating the ATE

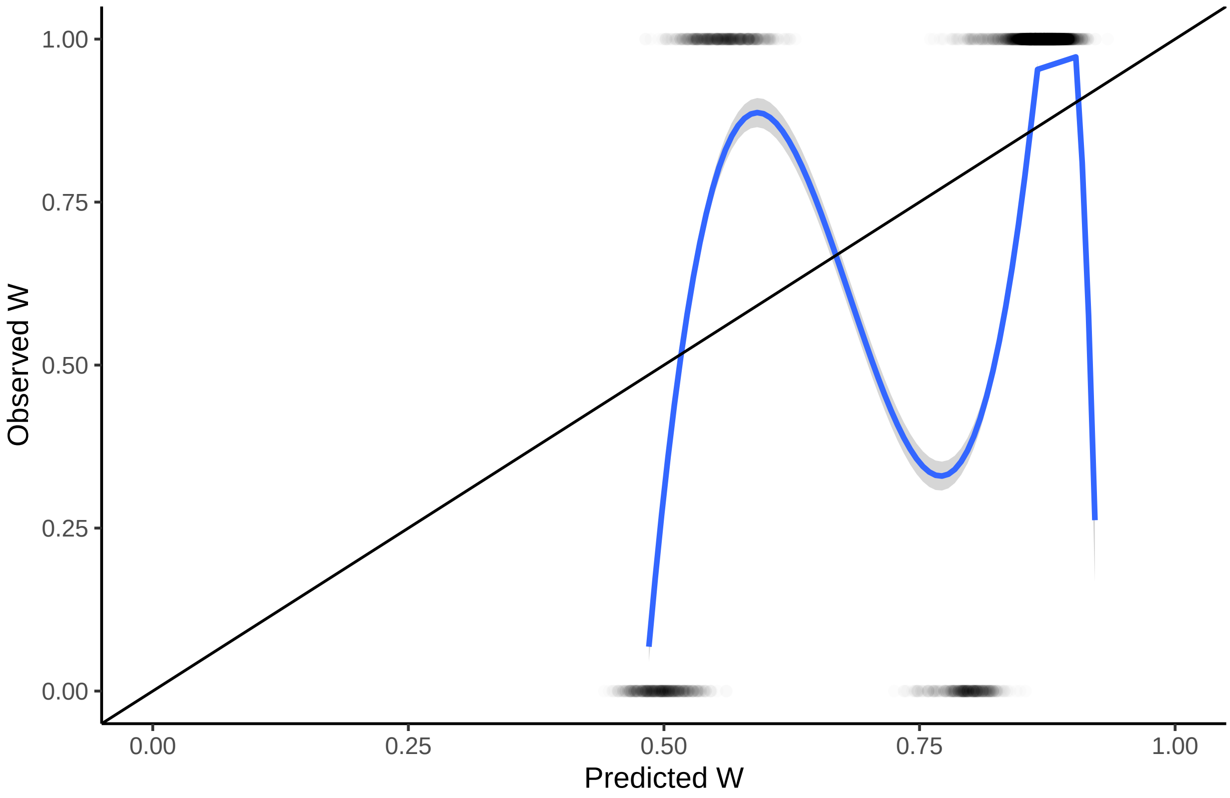
In this section we explore various methods for estimating the ATE. We explore the following methods:
1. inverse propensity weighting via logistic regression
2. direct regression analysis via OLS
3. traditional double robust analysis via augmented inverse-propensity score weighting that combines the above two estimators.
We also re-run the above methods after expanding the data to include all interactions of all of the covariates, and re-estimate outcome and proensity models using the original linear model, as well as running lasso and random forest models on the expanded data.

We first plot propensity scores against treatment status on the original and expanded set of coefficients. Models closer to the 45-degree line are better.
We see that logistic propensity scores perform surprisingly well!

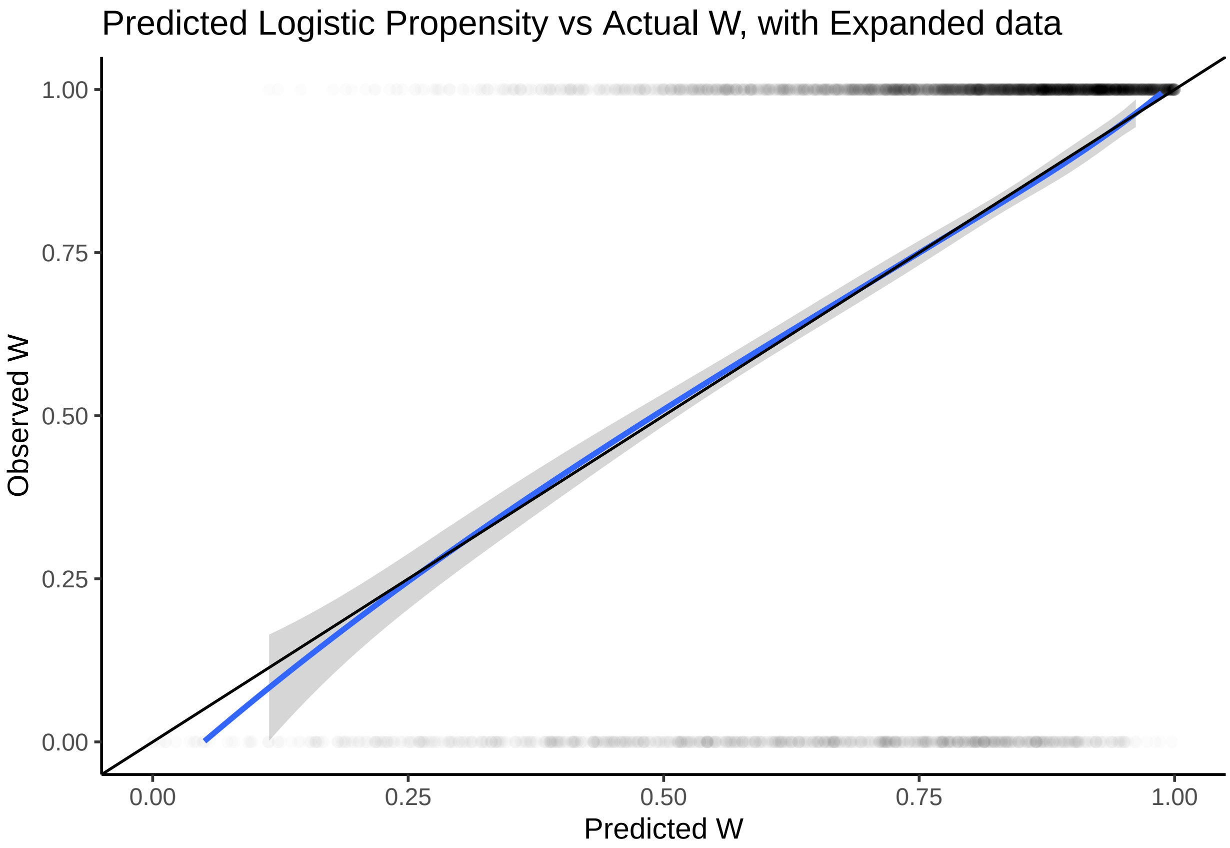


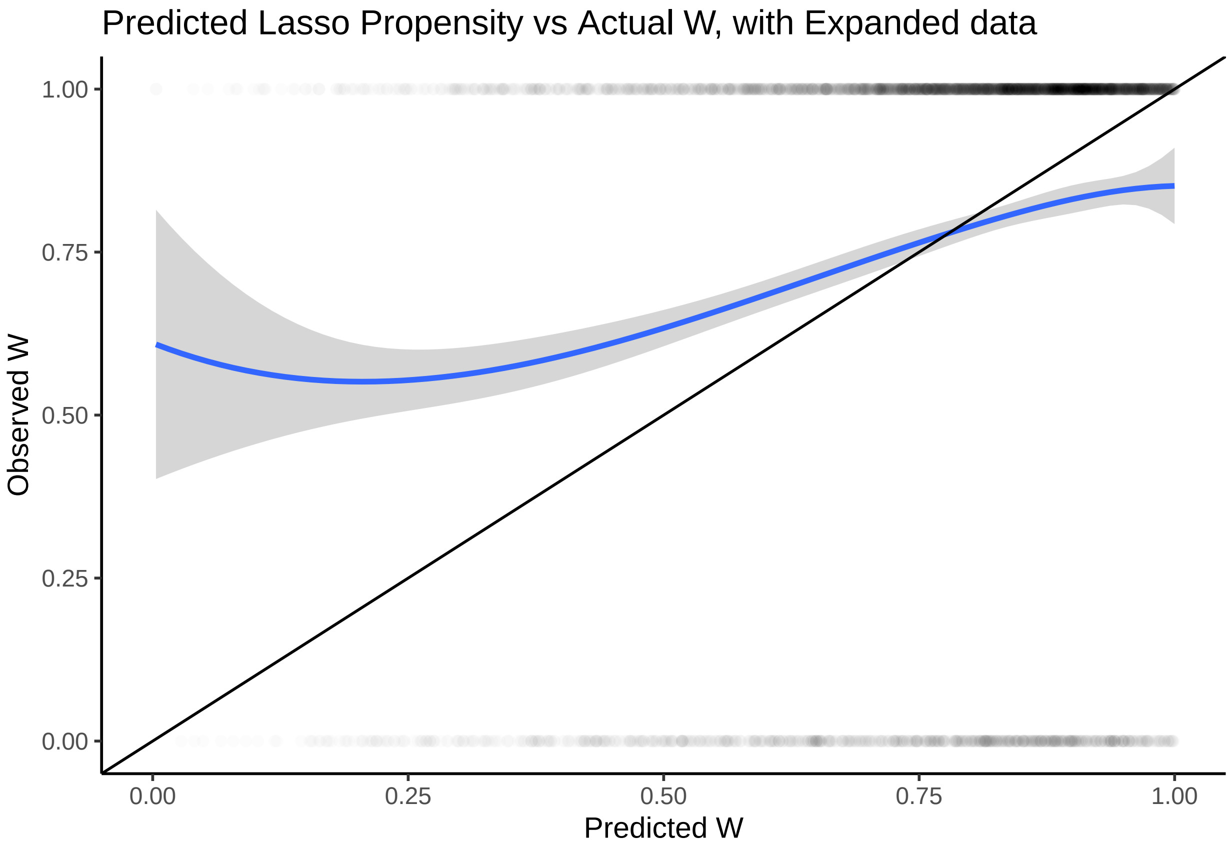


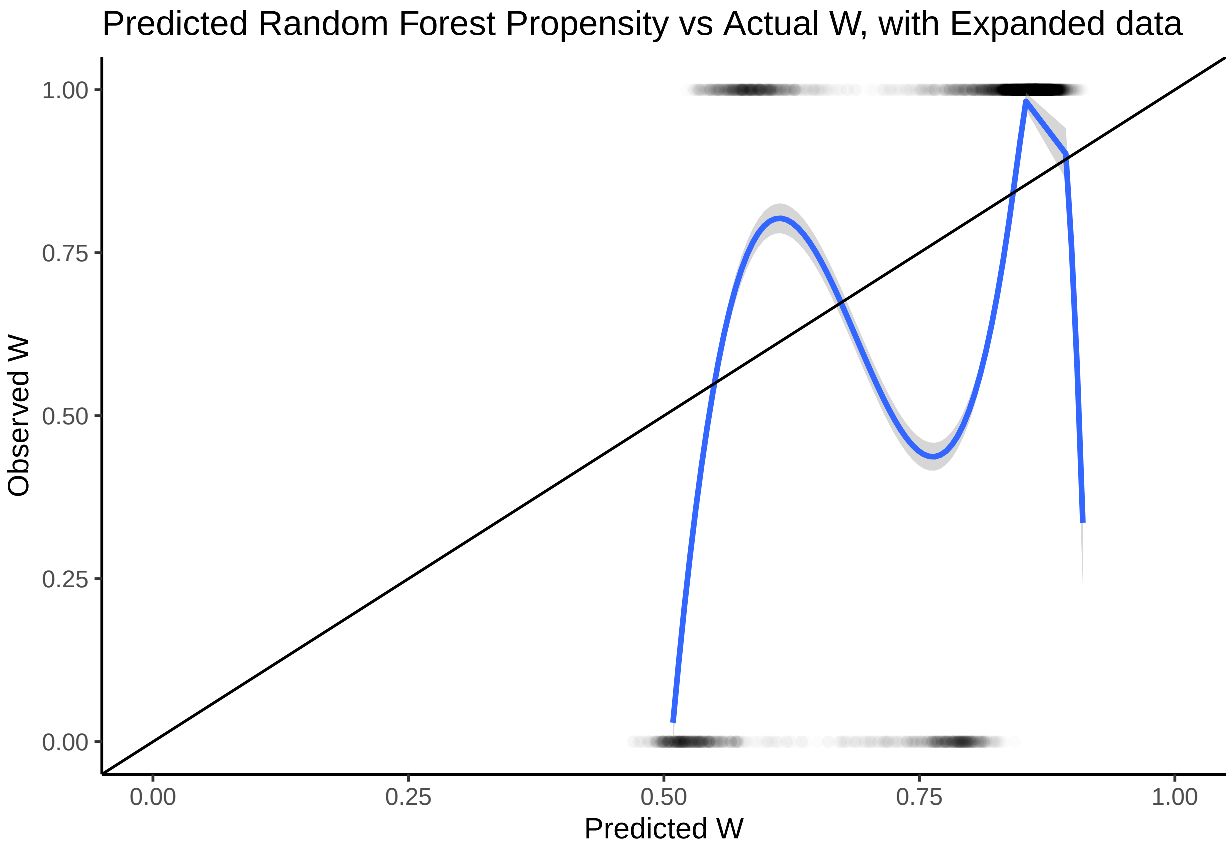
Predicted Lasso Propensity vs Actual W



Predicted Random Forest Propensity vs Actual W

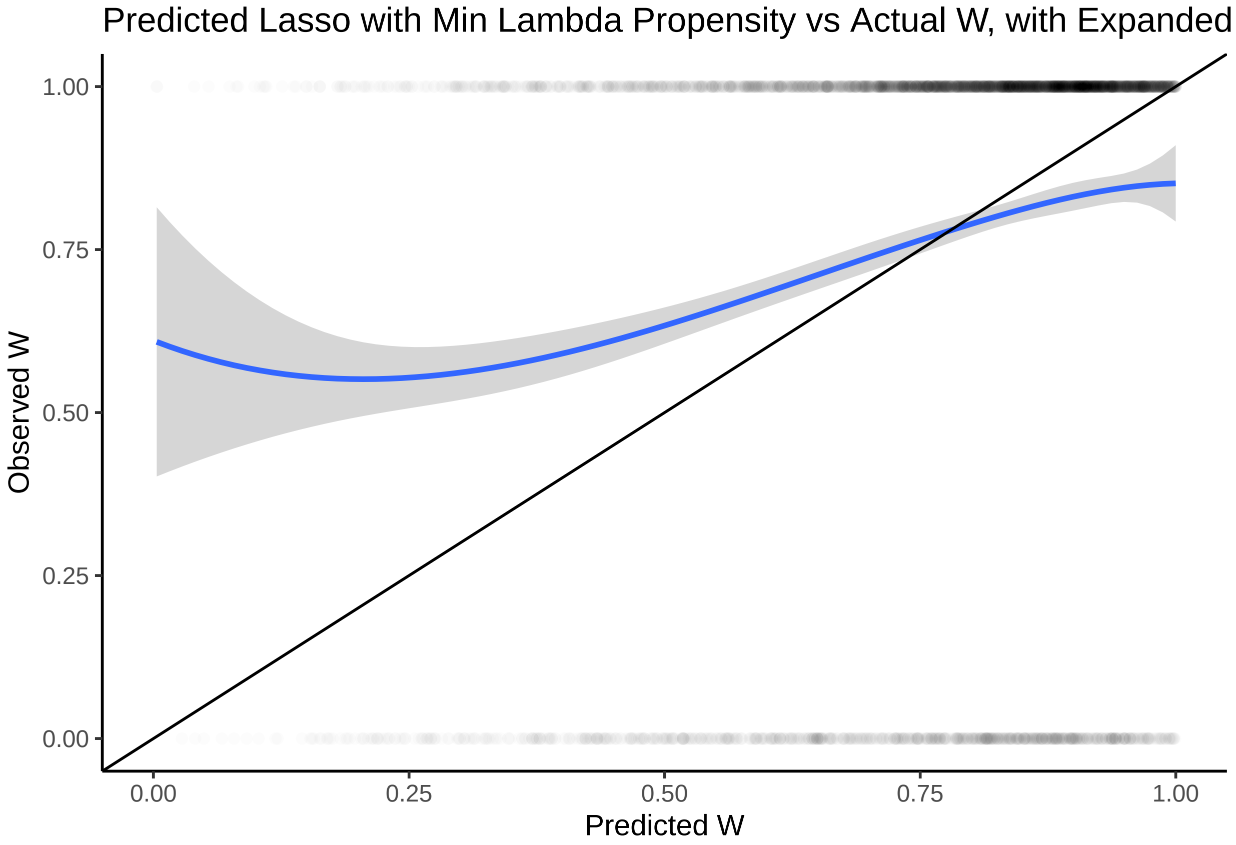

Predicted Logistic Propensity vs Actual W, with Expanded data

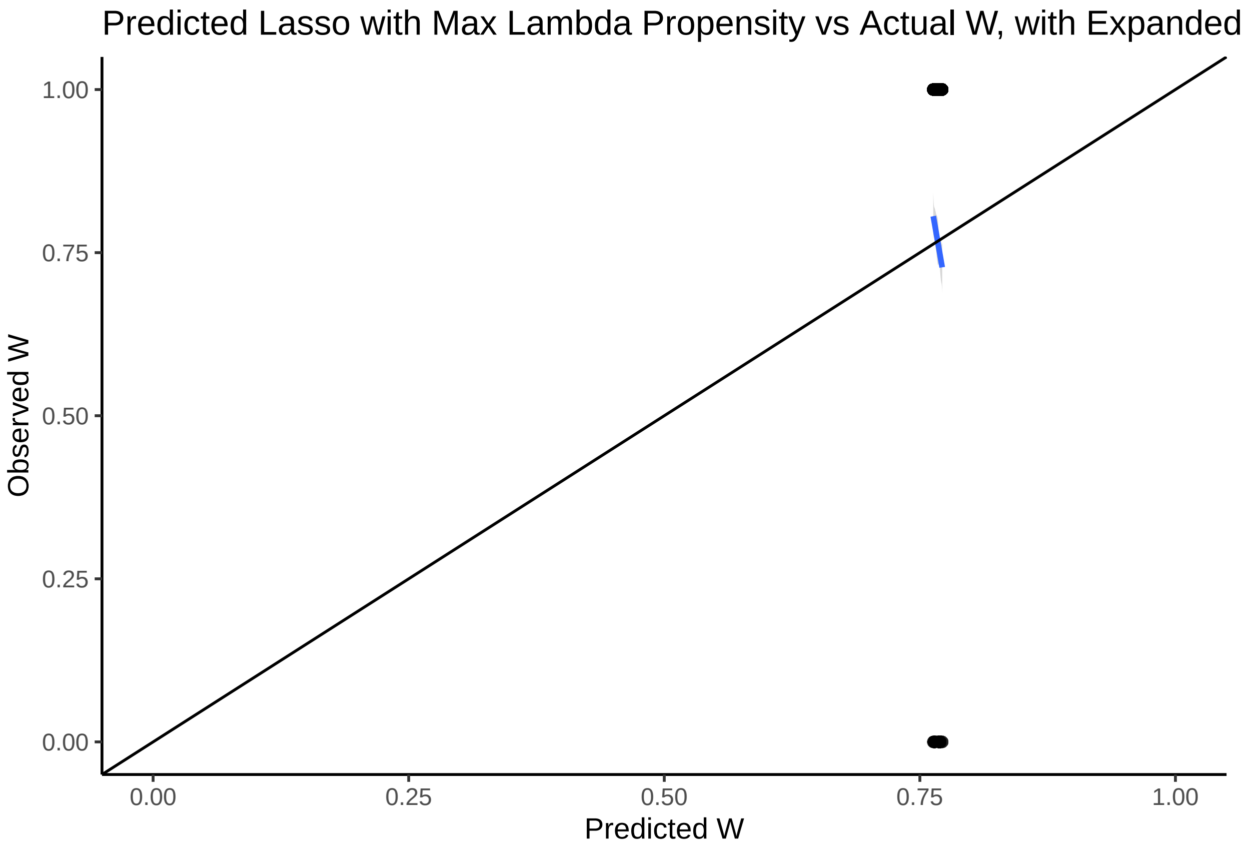

Predicted Lasso Propensity vs Actual W, with Expanded data


Predicted Random Forest Propensity vs Actual W, with Expanded data
Observed W
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
Predicted W

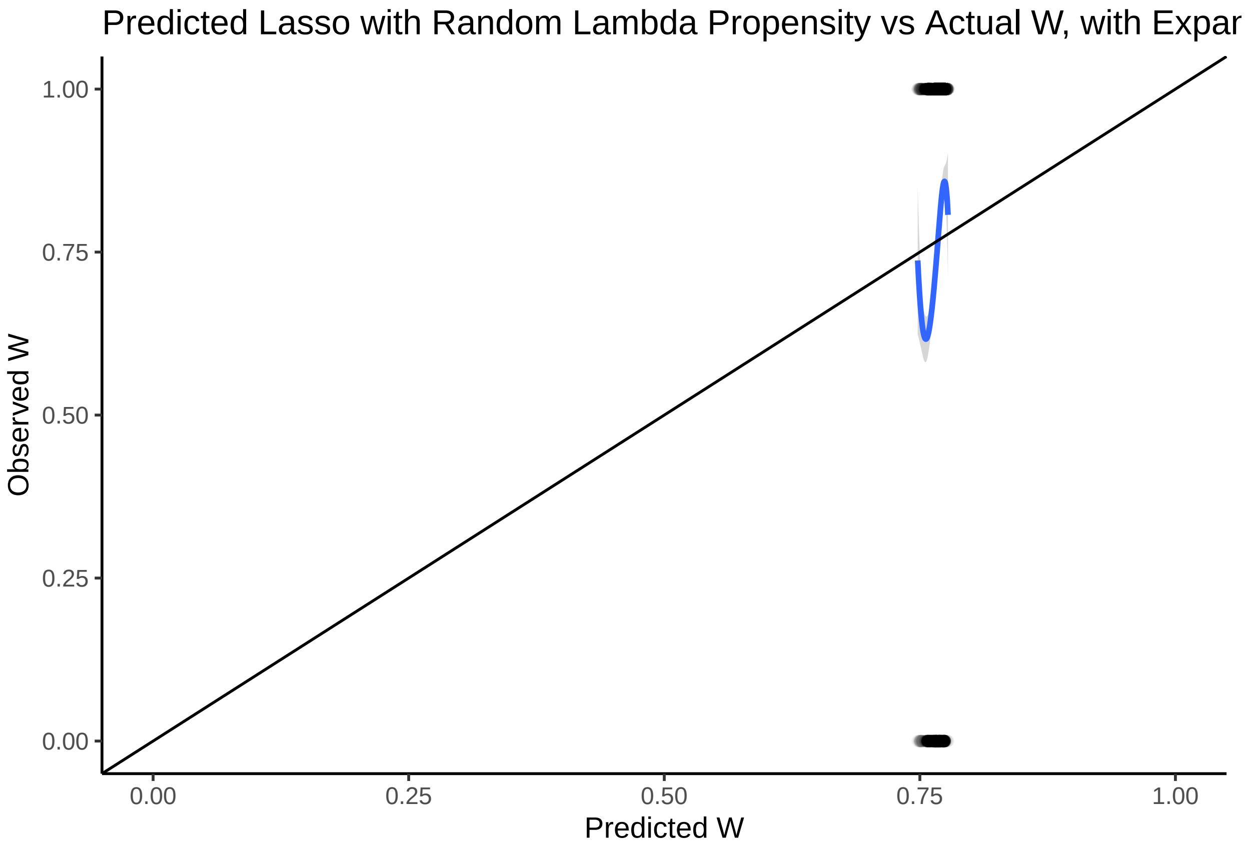
Exploring the Lasso Model Along Lambda

To show how cross-validating lambda is important for the lasso, we compare predicted and actual treatment status for the minimum, maximum, a randomly selected lambda. The lasso with the best lamda is the one closest to the 45-degree line.



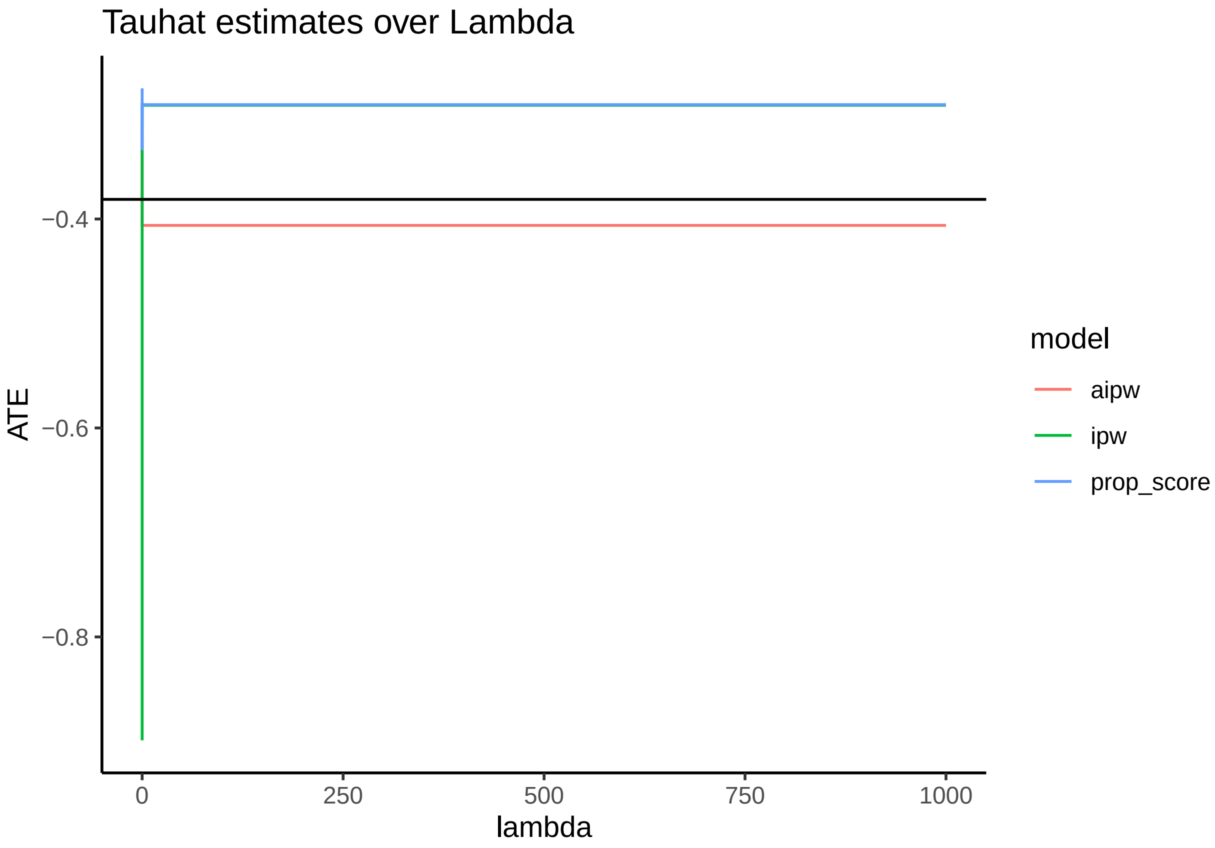


Predicted Lasso with Max Lambda Propensity vs Actual W, with Expanded

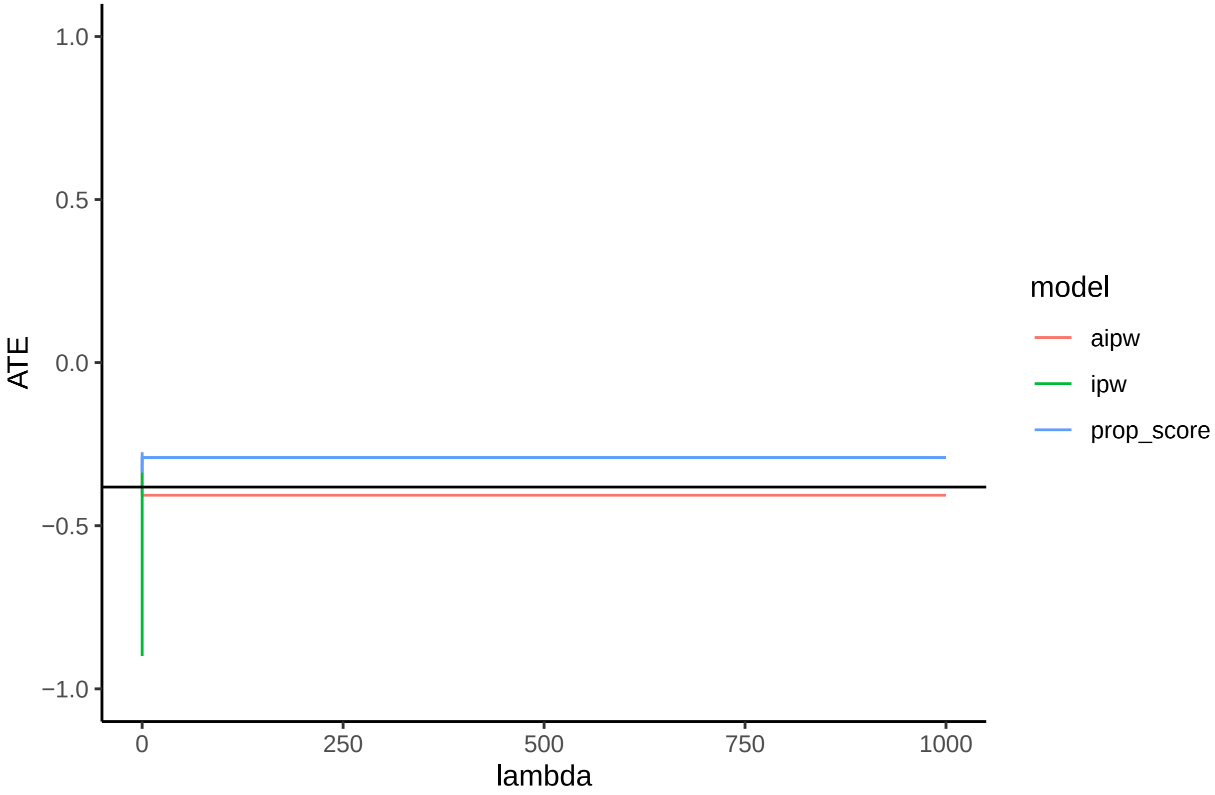


Predicted Lasso with Random Lambda Propensity vs Actual W, with Expar

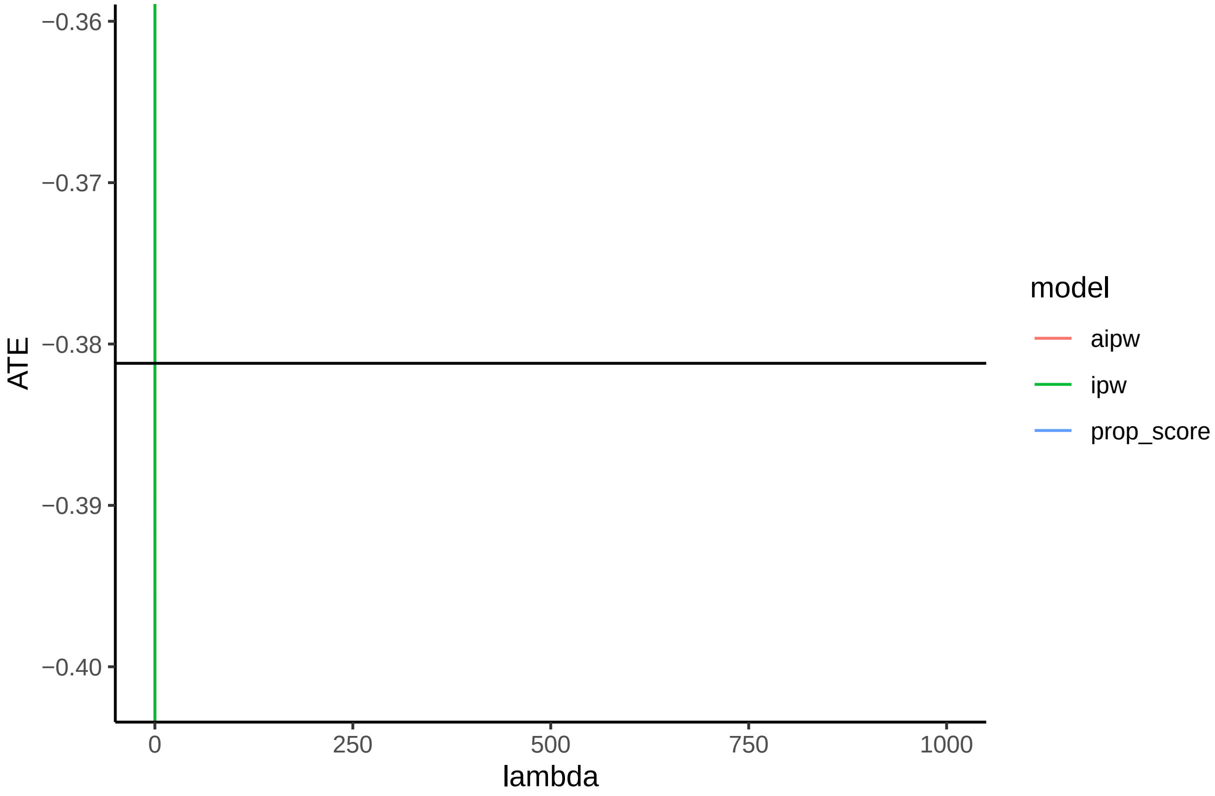
We also plot our various estimates of $\hat{\tau}$ over our lambdas.



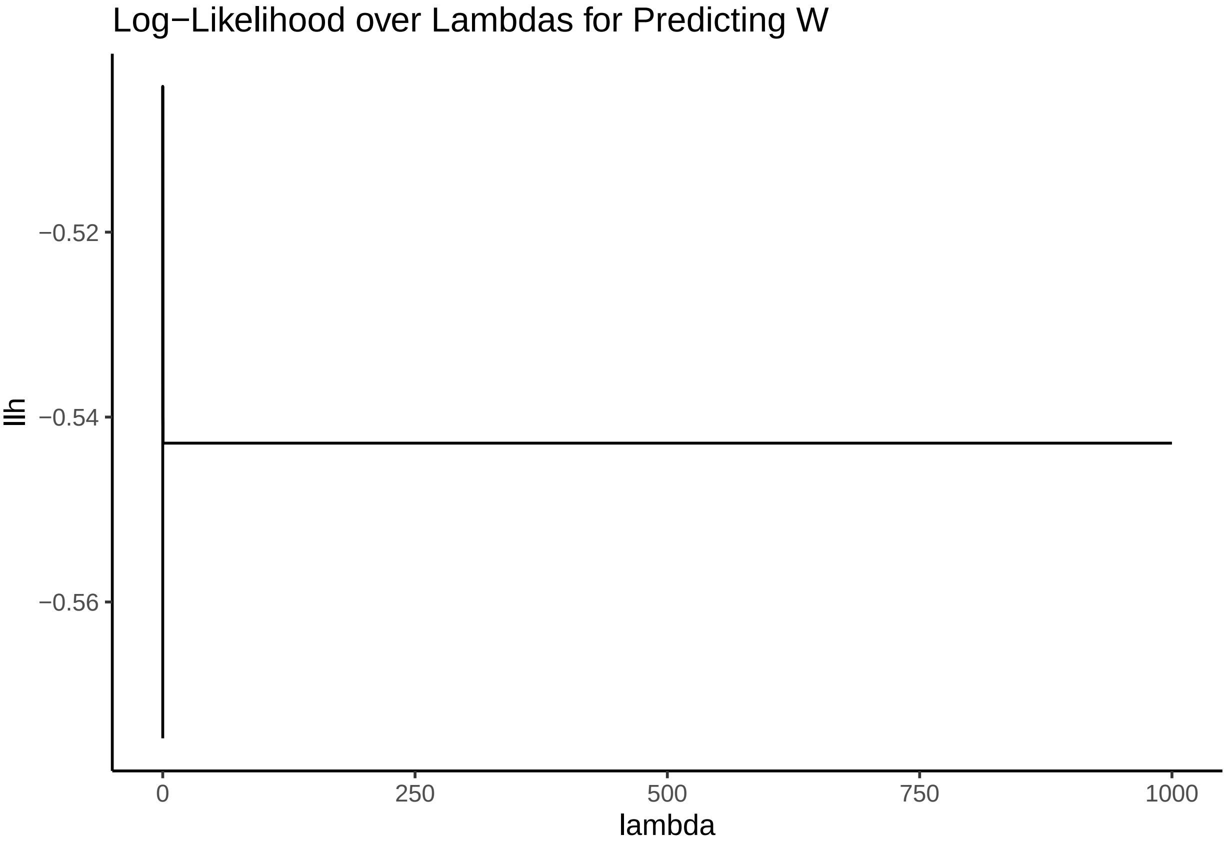
Tauhat estimates over Lambdas, zoomed in

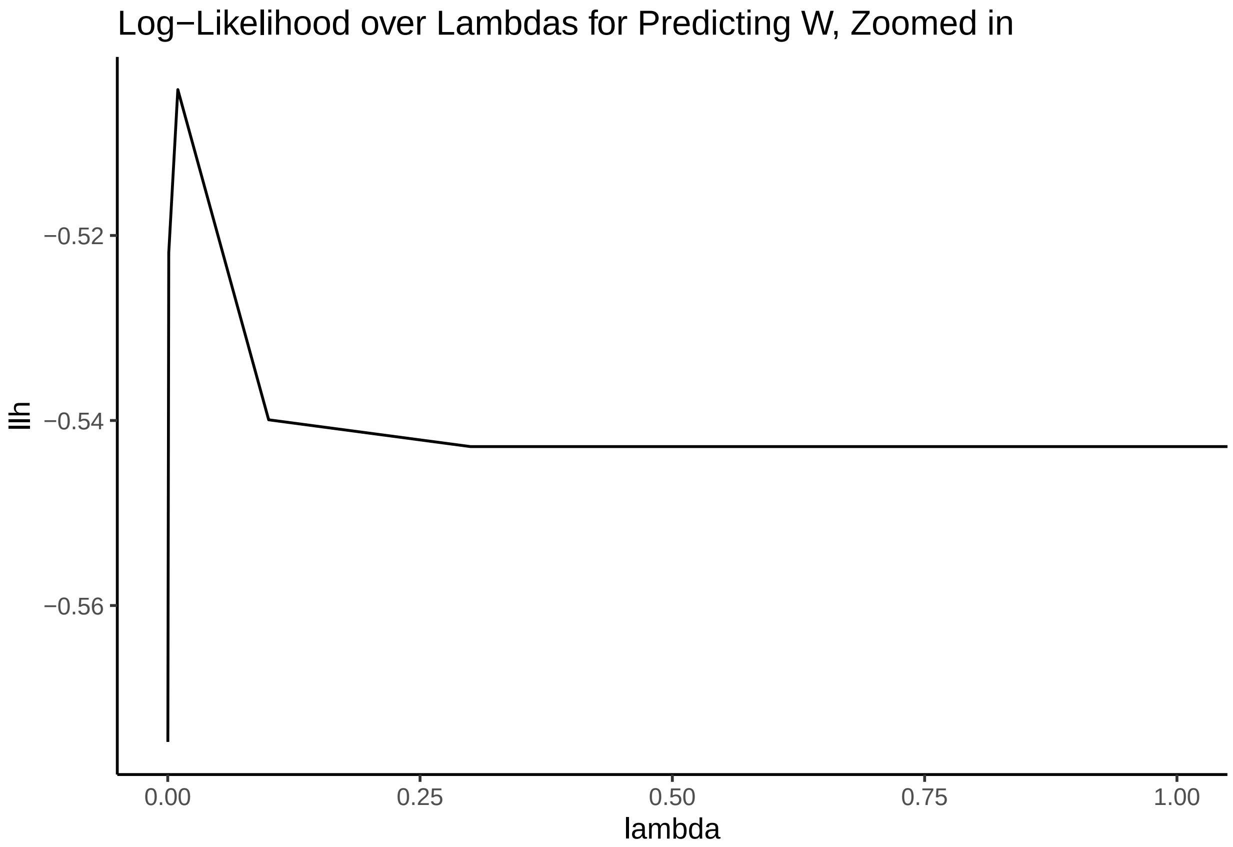


Tauhat estimates over Lambdas, zoomed in more



We now plot log likelihood over different values of lambda on the expanded data.





Comparing ATE Across Models with Original Data

Finally, we compare ATE across various models. We see that AIPW forest methods performs the best across the original and interacted data, though propensity weighted regression performed on the original data.

```
##                                    ATE lower_ci upper_ci ci_length
## RCT_gold_standard               -0.381   -0.401   -0.361     0.040
## naive_observational             -0.291   -0.311   -0.271     0.040
## linear_regression               -0.336   -0.372   -0.301     0.071
## propensity_weighted_regression -0.340   -0.377   -0.303     0.074
## IPW_logistic                    -0.351   -0.401   -0.301     0.100
## AIPW_linear_plus_logistic       -0.336   -0.372   -0.299     0.073
## IPW_forest                      -0.239   -0.275   -0.202     0.072
## AIPW_forest                     -0.343   -0.381   -0.306     0.075
## AIPW_linear_plus_forest         -0.342   -0.369   -0.316     0.054
## IPW_lasso                       -0.366   -0.418   -0.314     0.104
```

Comparing ATE Across Models with Interacted Data

```
##                                    ATE lower_ci upper_ci ci_length
## RCT_gold_standard               -0.381   -0.401   -0.361     0.040
## naive_observational             -0.291   -0.311   -0.271     0.040
## linear_regression               -0.406      NaN      NaN       NaN
## propensity_weighted_regression -0.398   -0.487   -0.309     0.179
## IPW_logistic                    -0.411   -0.571   -0.251     0.320
## AIPW_linear_plus_logistic      -0.410   -0.444   -0.376     0.069
## IPW_forest                     -0.225   -0.259   -0.191     0.069
## AIPW_forest                    -0.340   -0.376   -0.304     0.072
## AIPW_linear_plus_forest        -0.404   -0.431   -0.378     0.053
## IPW_lasso                      -0.291   -0.330   -0.252     0.078
```

Part Two

Justification of Propensity Stratification

Propensity stratification follows the same principle as stratified random experiments, where we would assign treatment after breaking people into groups based on their observable characteristics. This would ensure balance between treated and control units within strata (it would allow us to avoid overlap problems if done correctly). Propensity-based stratification functions similarly, as we are only comparing units with similar observable characteristics. Propensity scores provide a single-dimensional, unified measure with which to compare units. By comparing only “similar” units, we can ensure that our estimate of the treatment effect should be more accurate. By averaging our predictions over these strata, we can reduce the effect of bias present in only parts of the covariate space. When we increase the number of strata (fixing N), we narrow our comparison to more similar units, and reduce the effect of any poorly-estimated strata.

Propensity Stratification Function

Here we present a function to calculate propensity scores at a strata-level. The user supplies either a dataframe with treatment column W and outcome column Y, as well as a model with which to estimate propensity scores (we don’t supply a pre-calculated set of propensity scores in case we want to examine the usefulness of propensity stratification for new data). The function takes options for a number of strata and the function to estimate on the strata - the user could supply something more complex than the simple difference-in-means, for example.

The function checks that each strata has both treatment groups; if it does not then it is not included in the ATE calculation.

We fix the number of strata at 10, following the heuristic discussed in the homework.

Note that the true effect is `mean(W * X[,2])`.

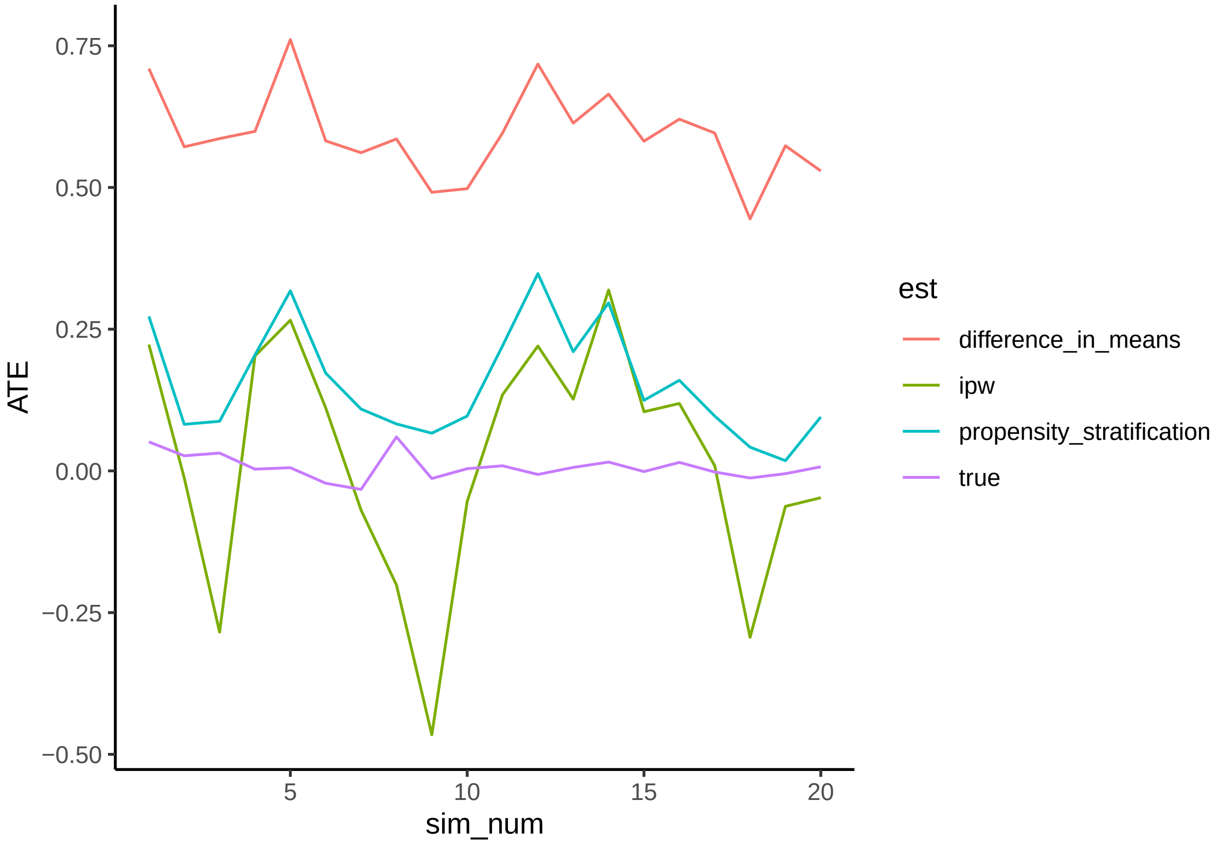
```
propensity_stratification <- function(df, treatment_model, n_strata = 10,
                                     tau_estimator = difference_in_means){
  df <- copy(df)
  df[, pW := predict(treatment_model, newdata = df[,.SD, .SDcols = !c('W', 'Y')], type = "response")]
  df[, pW_strata := ntile(pW, n_strata)]
  # if any strata is empty, automatically ignored
  # if all
  strata_count <- df[, .N, by = .(pW_strata, W)]
  strata_to_keep <- strata_count[, .(n_strata_W_nonempty = sum(N > 0)), by = .(pW_strata)][
    # keep strata if has observations in both treatment
    n_strata_W_nonempty == 2, pW_strata] %>% unique

  # sloppy but can't figure out right solution atm
  strata_tau_est <- df[pW_strata %in% strata_to_keep, .(
    tauhat = tau_estimator(.SD)[1],
    lower_ci = tau_estimator(.SD)[2],
    upper_ci = tau_estimator(.SD)[3]), by = .(pW_strata)][
      ,.(tauhat = mean(tauhat),
        lower_ci = mean(lower_ci),
        upper_ci = mean(upper_ci))]
  return(c(ATE = strata_tau_est$tauhat,
           lower_ci = strata_tau_est$lower_ci,
           upper_ci = strata_tau_est$upper_ci))
}

#### SIMULATION ####
```

Simulation Exercise

For all discussion that follows, we use the following code to generate a simulated dataset:

```
make_simulation <- function(){
  n = 1000; p = 20
  X = matrix(rnorm(n * p), n, p)
  propensity = pmax(0.2, pmin(0.8, 0.5 + X[,1]/3))
  W = rbinom(n, 1, propensity)
  Y = pmax(X[,1] + W * X[,2], 0) + rnorm(n)
  df = cbind(X, W, Y) %>% as.data.frame() %>% setDT
  df
}
```

We now run 20 simulations of the type described above, and report results over each simulation.
We see that propensity stratification and IPW perform similarly, but propensity stratification has lower average MSE.


0.75
0.50
0.25
0.00
−0.25
−0.50
ATE
5
10
15
20
sim_num
est
difference_in_means
ipw
propensity_stratification
true