final_project.R

mlinegar

2020-06-03

Contents

0.1 RCT Analysis

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

```
tauhat_rct <- difference_in_means(df)
print(tauhat_rct)
```

```
##      ATE lower_ci upper_ci
##  -0.0412  -0.0708  -0.0117
```

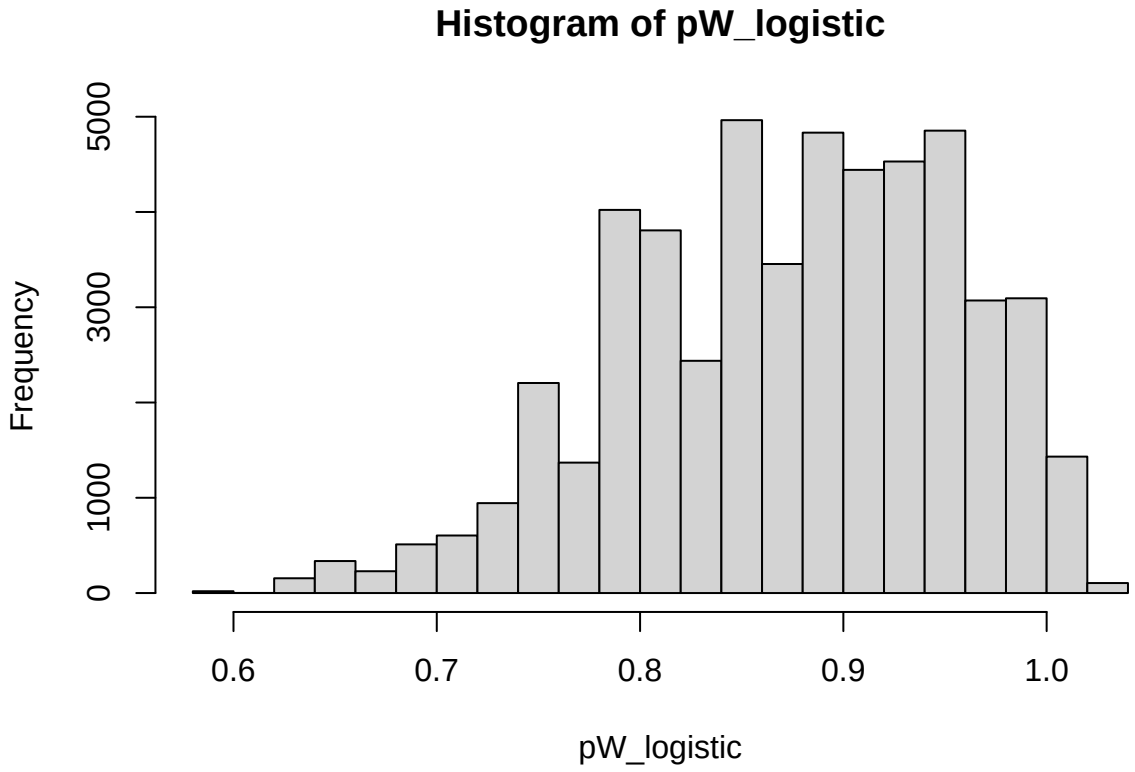
```
#### TESTING ASSUMPTIONS ####
```

0.2 Testing Assumptions

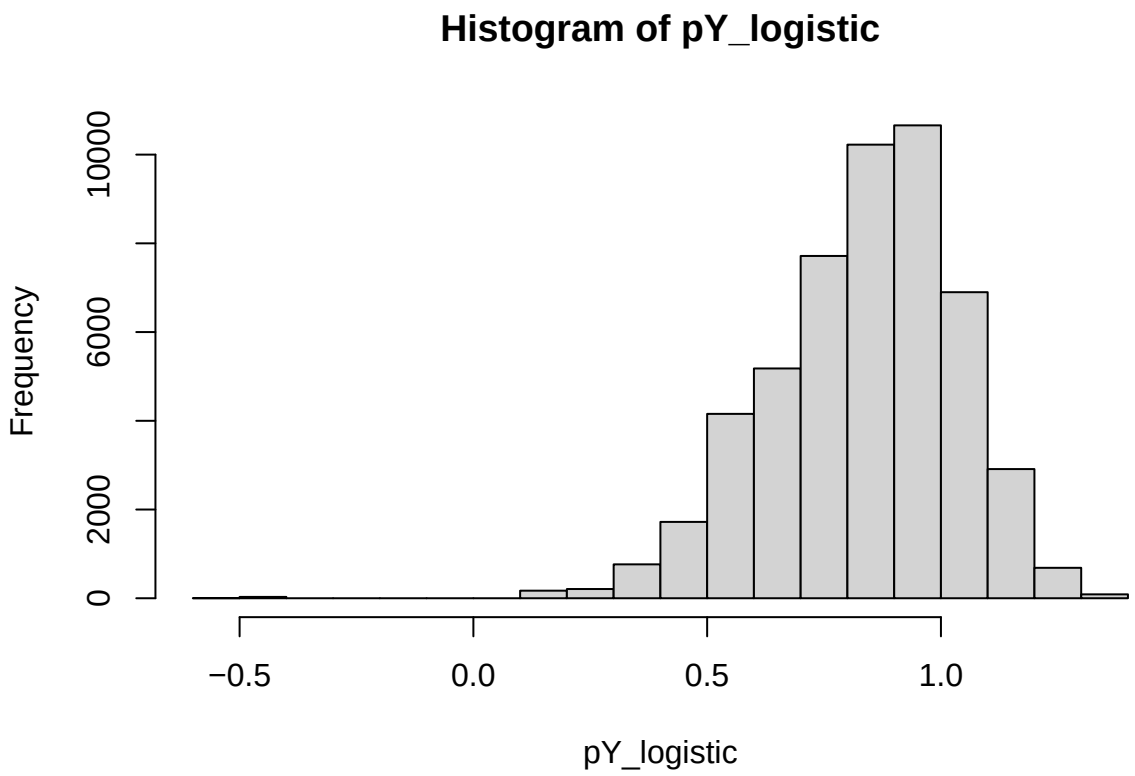
Here we test some of our traditional causal inference assumptions.

As a first step, we plot logistic predictions of the probabilities our treatment pW and our outcome pY (which is binary). We see that treatment assignment appears to follow a normal distribution, and that our outcome has an average unconditional probability of 0.835.

```
pW_logistic.fit <- glm(Wmod ~ as.matrix(Xmod), family = outcome_family)
pW_logistic <- predict(pW_logistic.fit, type = "response")
pW_logistic.fit.tidy <-  pW_logistic.fit %>% tidy()
hist(pW_logistic)
```




```
pY_logistic.fit <- glm(Ymod ~ as.matrix(Xmod), family = outcome_family)
pY_logistic <- predict(pY_logistic.fit, type = "response")
pY_logistic.fit.tidy <-  pY_logistic.fit %>% tidy()
hist(pY_logistic)
```




```
df_mod[, `:=`(p_Y = pY_logistic,
              p_W = pW_logistic)]
```

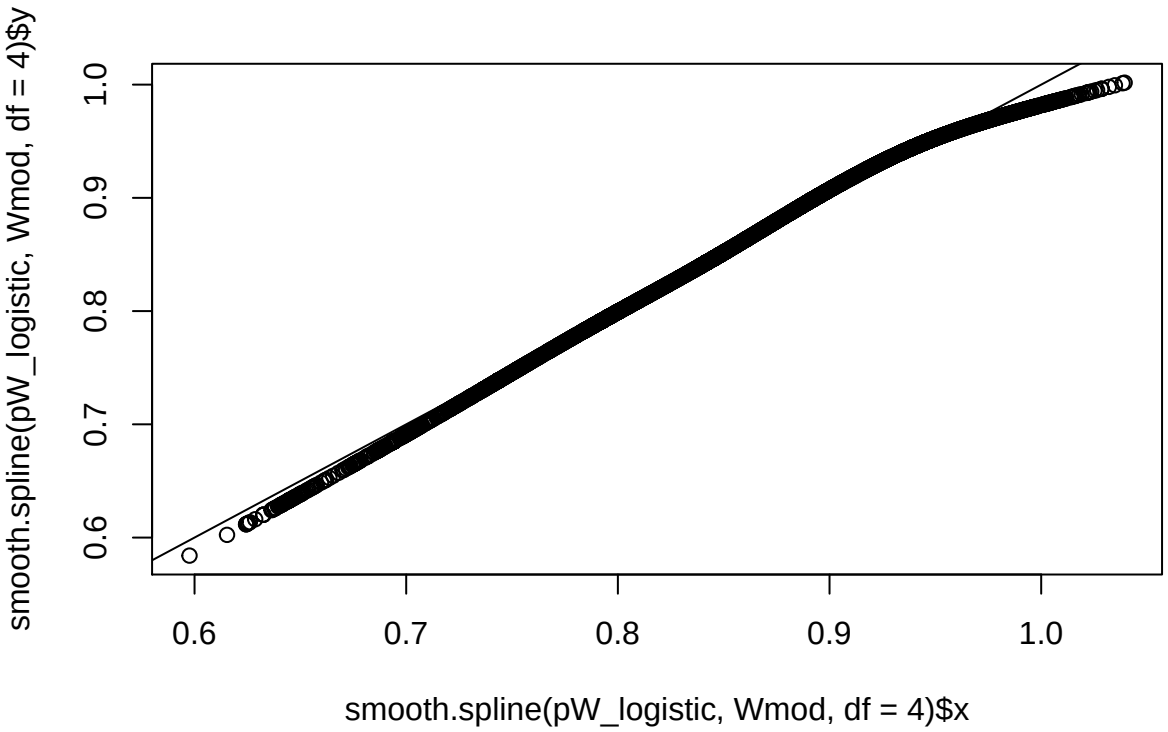
Some summary statistics of our data:

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.

```
{plot(smooth.spline(pW_logistic, Wmod, df = 4))
  abline(0, 1)}
```

Table 1:

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
age	51,419	6.370	1.370	2	6	7	12
grade_level	51,419	4.180	1.580	1	3	5	7
book_read	51,419	0.682	3.940	0	0	0	120
book_added	51,419	1.040	4.110	0	0	1	119
score	51,419	5.660	4.380	0	0	10	10
child_number	51,419	1.100	0.353	1	1	1	3
total_session_time_in_minutes	51,419	10,119.000	19,228.000	0.000	582.000	10,520.000	194,196.0
story_read_time_in_min	51,419	851.000	2,237.000	0	77.2	933.0	51,350
qa_accuracy	51,419	70.000	22.300	0.000	58.200	86.800	100.000
story_qa_accuracy	51,419	68.000	26.300	0.000	55.600	87.600	100.000
freadom_point	51,419	182,963.000	261,810.000	0	22,700	273,360	5,403,09
commitment_taken	51,419	0.701	0.458	0	0	1	1
completed_activities	51,419	63.800	56.700	0	6	126	254
completed_packs	51,419	9.030	8.800	0	0	19	36
completed_pack_activities	51,419	5.420	7.130	0	0	12	69
bookmarked_story	51,419	32.100	104.000	0	0	9	847
news_fread_completed	51,419	142.000	169.000	0	0	235	820
total_questions	51,419	767.000	676.000	0	115	1,302	3,080
activity_qa_accuracy	51,419	62.200	26.900	0.000	49.500	81.900	100.000
news_fread_qa_accuracy	51,419	60.800	40.000	0	0	94.2	100
reading_qa_accuracy	51,419	60.500	33.500	0.000	40.000	88.000	100.000
freadom_points_earned	51,419	180,524.000	260,290.000	0	22,100	266,950	5,397,35
num_parental_commitment	51,419	1.610	1.250	0	0	3	3
num_interests	51,419	2.050	4.080	0	0	1	17
W	51,419	0.874	0.332	0	1	1	1
Y	51,419	0.835	1.130	0	0	1	3
p_Y	51,419	0.835	0.199	−0.508	0.713	0.974	1.350
p_W	51,419	0.874	0.081	0.598	0.812	0.939	1.040




```
#### RCT ANALYSIS ####
```

0.3 RCT Analysis

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

```
tauhat_rct <- difference_in_means(df)
print(tauhat_rct)
```

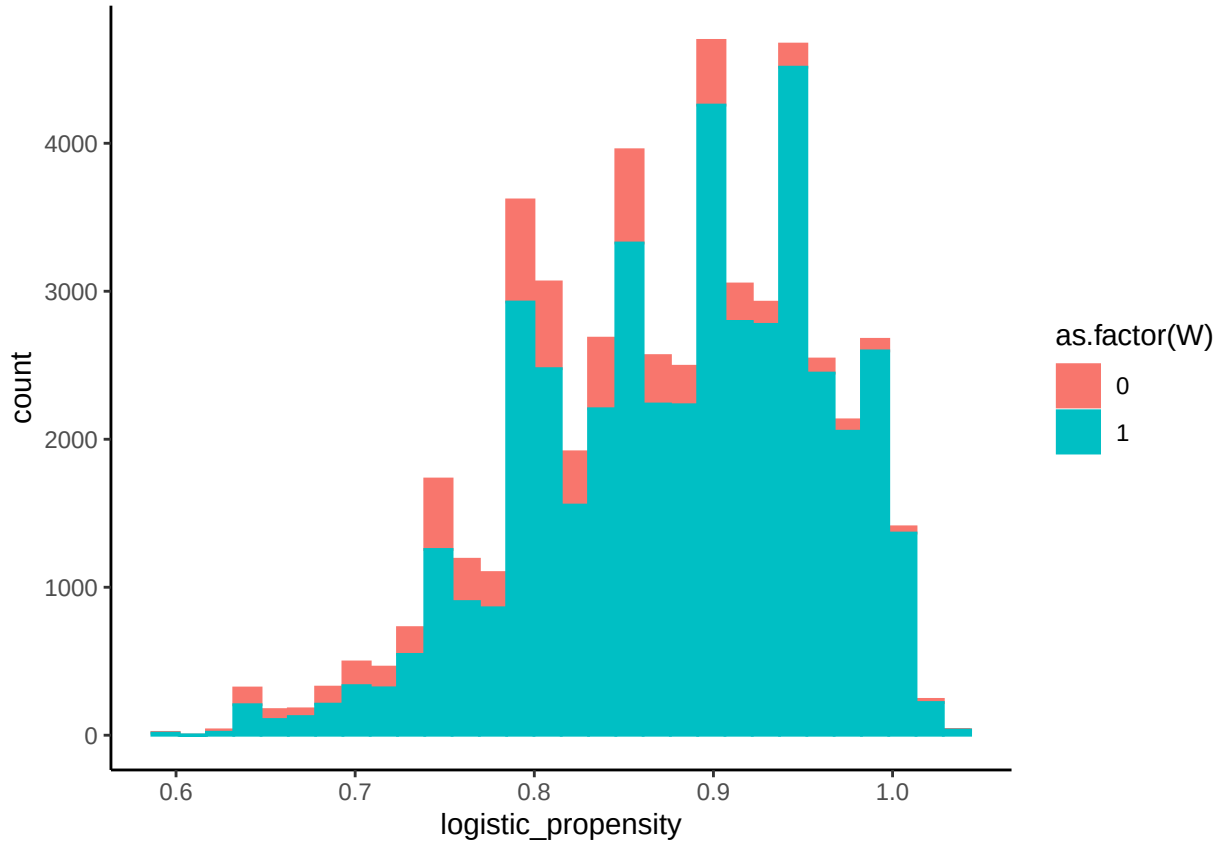
```
##      ATE lower_ci upper_ci
##  -0.0412  -0.0708  -0.0117
```

```
#### LOGISTIC PROPENSITY SCORES ####
```

```
# df_mod <- copy(df)
Xmod = df_mod[,.SD, .SDcols = names(df_mod)[!names(df_mod) %in% c("Y", "W")]] %>% as.matrix()
Ymod = df_mod$Y
Wmod = df_mod$W
XWmod = cbind(Xmod, Wmod)

# Computing the propensity score by logistic regression of W on X.
pW_logistic.fit <- glm(Wmod ~ as.matrix(Xmod), family = outcome_family)
pW_logistic <- predict(pW_logistic.fit, type = "response")

df_mod[, logistic_propensity := pW_logistic]

#### OVERLAP ####
```

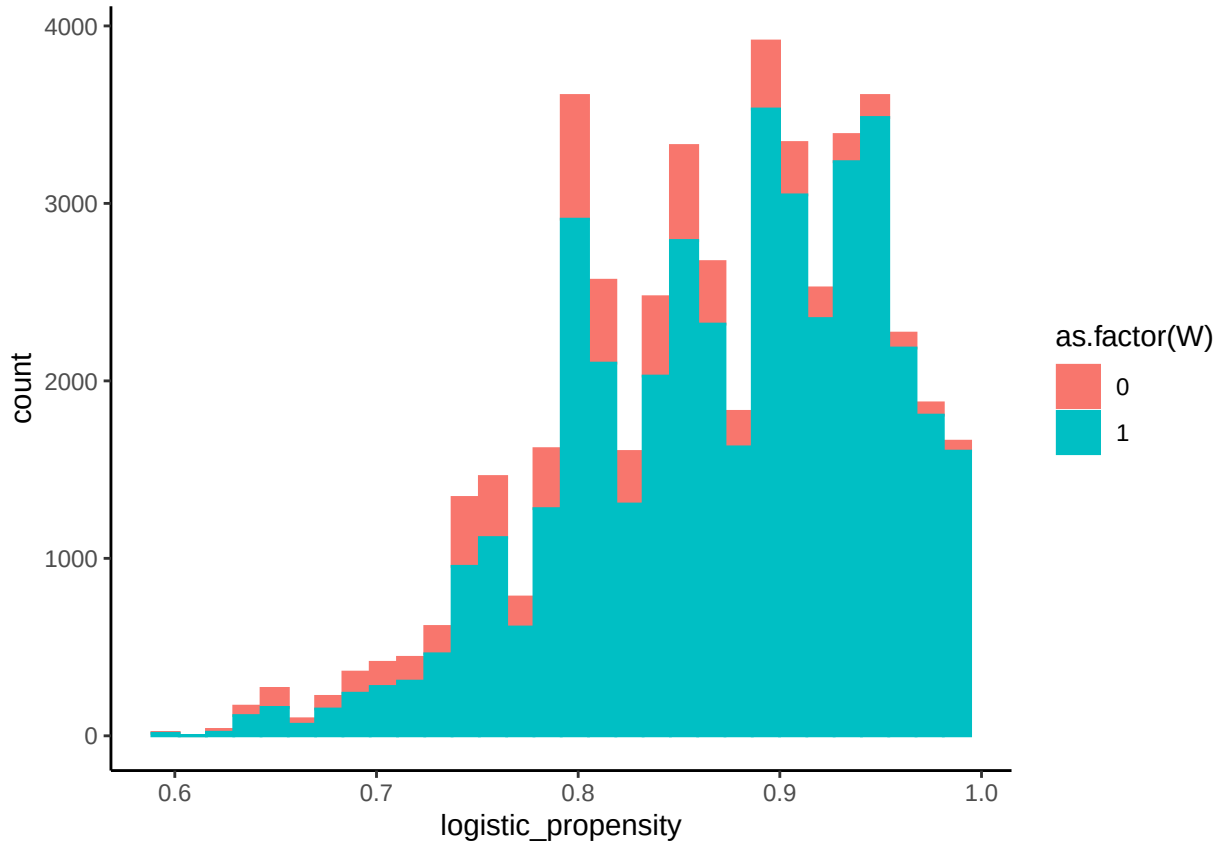
We now plot (logistic) propensity scores, showing that we still have overlap after removing observations. We may be somewhat concerned about the small number of observations with propensities close to one; so remove all observations with propensity score outside of 0.01 and 0.99 to fix this. We then re-estimate the propensity model.
We first show overlap before truncating.

```
overlap <- df_mod %>% ggplot(aes(x=logistic_propensity,color=as.factor(W),fill=as.factor(W)))+ geom_his
overlap
```



We now truncate, and plot truncated overlap.

```
df_mod <- df_mod[logistic_propensity %between% propensity_bound]

Xmod = df_mod[,.SD, .SDcols = names(df_mod)[!names(df_mod) %in% c("Y", "W")]] %>% as.matrix()
Ymod = df_mod$Y
Wmod = df_mod$W
XWmod = cbind(Xmod, Wmod)

overlap <- df_mod %>% ggplot(aes(x=logistic_propensity,color=as.factor(W),fill=as.factor(W)))+ geom_his
overlap
```

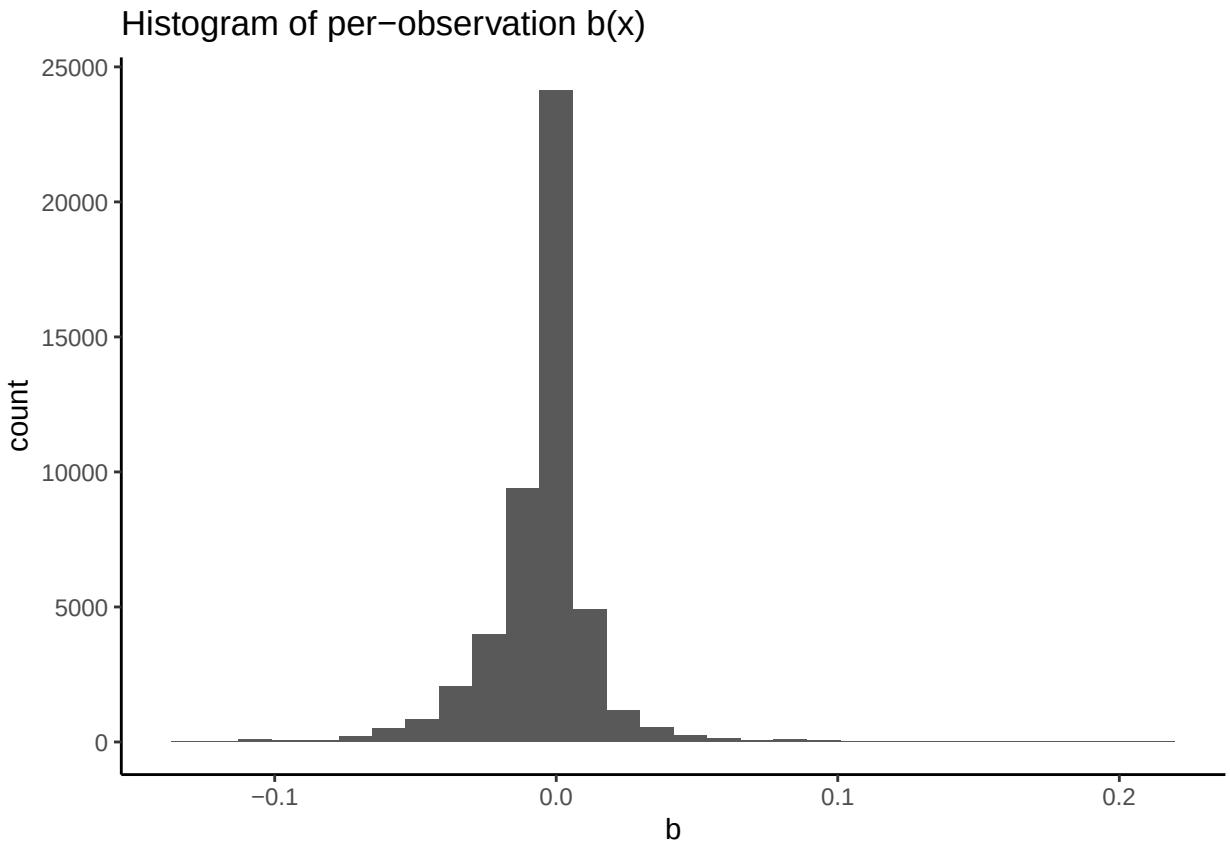



```
#### PREDICTING PROPENSITIES AND OUTCOMES, ORIGINAL AND EXPANDED DATA ####
# some of this is used to calculate bias function, hence the ordering

# logistic model
# original data
pW_logistic.fit <- glm(Wmod ~ Xmod, family = outcome_family)
pW_logistic <- predict(pW_logistic.fit, type = "response")

# original data
pY_logistic.fit <- glm(Ymod ~ XWmod, family = outcome_family)
pY_logistic <- predict(pY_logistic.fit, type = "response")

# lasso expanded data, code provided by TA
# original data
pW_glmnet.fit.model = glmnet::cv.glmnet(Xmod, Wmod, lambda = lambda, family = outcome_family, type.measu
pY_glmnet.fit.model = glmnet::cv.glmnet(Xmod, Ymod, lambda = lambda, family = outcome_family, type.measu
# expanded data

# demonstration of lasso fit across lambdas:
pW_lasso = pW_glmnet.fit.model$fit.preval[, pW_glmnet.fit.model$lambda == pW_glmnet.fit.model$lambda.mi
pW_lasso.min = pW_glmnet.fit.model$fit.preval[, pW_glmnet.fit.model$lambda == min(pW_glmnet.fit.model$la
pW_lasso.max = pW_glmnet.fit.model$fit.preval[, pW_glmnet.fit.model$lambda == max(pW_glmnet.fit.model$la
pW_lasso.rand = pW_glmnet.fit.model$fit.preval[, pW_glmnet.fit.model$lambda == base::sample(pW_glmnet.fi

# random forest
pW_rf.fit = regression_forest(Xmod, Wmod, num.trees = 500)
pY_rf.fit = regression_forest(Xmod, Ymod, num.trees = 500)
```

```
# pW_rf = pW_rf.fit$predictions
# pY_rf = pY_rf.fit$predictions
pW_rf = predict(pW_rf.fit, newdata = Xmod) %>% as.matrix
pY_rf = predict(pY_rf.fit, newdata = Xmod) %>% as.matrix

# CF
cf = causal_forest(Xmod, Ymod, Wmod, num.trees = 500)

#### BIAS FUNCTION ####
```

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.

```
mu_avg <- function(treated, df){df[W==treated, mean(Y)]}
mu <- function(treated, df){df[W==treated, mean(pY)]}

B <- function(df, treatment_model, outcome_model, outcome_type = "response"){
  # have to supply models so that can estimate counterfactual predictions given an alternative treatmen
  # note that this will NOT work for lasso model, attempt to warn of this misbehavior:
  if (grepl("lasso|rf|cf", deparse(quote(treatment_model)))){
    simpleMessage("The predict method appears to be broken for lasso models estimated using glmnet::cv.g
                  You may want to try another predictive model.")
  }
  df = copy(df)
  p = df[,mean(W)]
  mu0 <- df[W==0,mean(Y)]
  mu1 <- df[W==1,mean(Y)]

  pY_w0 <- predict(outcome_model, newdata = df[,.SD, .SDcols = !c('W', 'Y')][, W := 0], type = outcome_t
  pY_w1 <- predict(outcome_model, newdata = df[,.SD, .SDcols = !c('W', 'Y')][, W := 1], type = outcome_t
  pW <- predict(treatment_model, newdata = df[,.SD, .SDcols = !c('W', 'Y')], type = "response")
  df[, `:=`(W = NULL, pY_w0 = pY_w0, pY_w1 = pY_w1, pW = pW)]


  df[, b := (pW - p) * (p * (pY_w0 - mu0) + (1 - p) * (pY_w1 - mu1))]

  return(df[,.(b)])
}

df_mod_bias <- B(df_mod, pW_logistic.fit, pY_logistic.fit)
ggplot(df_mod_bias, aes(x = b)) + geom_histogram() + labs(title = "Histogram of per-observation b(x)")
```




```
#### ESTIMATING ATE INTRO ####
```

0.4 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.

0.5 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              -4.10e-02 -7.10e-02 -1.20e-02  5.90e-02
## naive_observational            -2.73e-01 -2.88e-01 -2.59e-01  2.90e-02
## linear_regression               9.00e-03 -2.50e-02  4.30e-02  6.80e-02
## propensity_weighted_regression  3.05e-01 -6.80e-02  6.79e-01  7.47e-01
## IPW_logistic                   -1.44e+10 -2.66e+10 -2.31e+09  2.43e+10
## AIPW_linear_plus_logistic      -5.79e+09 -1.68e+10  5.22e+09  2.20e+10
## IPW_forest                           NaN       NaN       NaN       NaN
## AIPW_forest                          NaN       NaN       NaN       NaN
## AIPW_linear_plus_forest              NaN       NaN       NaN       NaN
## IPW_lasso                     6.59e-01  6.37e-01  6.81e-01  4.30e-02
```