An Exploration of ATE and HTE: The Effects of an Environmental Tax on Voting Patterns

Patricio Ortiz
Department of Economics
patricio.ortiz@stanford.edu

Diane Sarkis
Department of Economics
diane.sarkis@stanford.edu

Andrew Conkey
Department of Economics
aconkey@stanford.edu

Krista Arenaodu
Department of Economics
arenaodu@stanford.edu

1 Motivation

While governments in many countries increasingly view environmental protection as a central priority, advancing it can be politically challenging. Despite the distant rewards of such policies, they pose immediate costs that are often unevenly distributed and more heavily placed on the less well off. Recent literature conjectures that the backlash instigated by green policies is particularly well aligned with the right-wing populist agenda, who are prone to taking a skeptical stance on environmental issues and in doing so expect to attract voters discontented with the burden of environmental taxes. However, the extent to which the introduction of green policies affects vote choice remains an open question. Our study replicates the data from one of the latest literature on this topic, "The Political Consequences of Green Policies: Evidence from Italy" by Colantone et al. [1], to examine whether a costly car ban in the city of Milan, issued by the social democratic party, led to an electoral shift among car owners impacted by the ban. First advanced in July 2019, the policy restricted certain polluting vehicle models from circulating within a large area that covers over 70 percent of Milan and where 97 percent of the city population resides. Most vocally opposed to the policy were representatives of the far-right populist party Lega, who attacked the environmental policy approach of the Democratic Party on the grounds that such initiative would "depress the economy and penalize the weaker social segments."

By exploiting arbitrary discontinuities in the rules dictating the models that would be covered by the ban, the original paper by Colantone at el. employs a difference-in-difference approach, revealing that owners of banned cars were 13.5 percentage points more likely to vote for Lega in the European Parliament elections of 2019. They use the following figure to visualize the difference-in-difference approach:

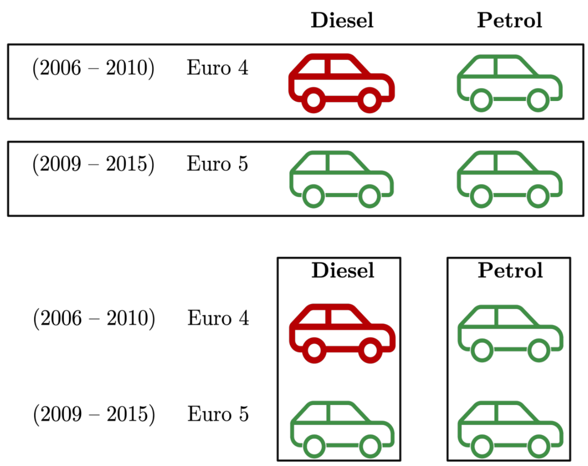


Figure 1: Colantone et al. Diff-in-Diff Approach

The estimated electoral shift from the original paper appears extraordinarily large, thus providing motivation for us to validate the average treatment effect of the Area B Policy using ML methods. It is fairly implausible that affected voters would be so motivated to change their vote by a single policy, especially given that their car values should not have been substantially affected by the policy—they could always sell their car to someone living outside Milan. Taking into account of the distributional concerns related to green policies, we believe it is also important to examine whether the car ban impacts some group more than others with respect to voting decision. Therefore, we expand on the scope of inquiry of the paper to check for heterogeneous treatment effect.

2 Data and General Methodology

2.1 Covariates

We limit ourselves to a subset of the covariates collected by the original survey, in part to limit overfitting given that the dataset is relatively small and in part to avoid colinearity issues. We make use of general personal information such as age, gender, and education level, and also make us of data capturing the nature and frequency of participants' car use, including how often they use their car and how many kilometers they typically drive it. Alongside these, we use survey responses intended to capture respondents' attitudes toward environmental issues. The include questions on whether they believe firms or governments should be responsible for the environment, whether they support taxes for environmental causes, whether they generally support green policies, whether they support certain carbon neutrality goals, whether they are willing to pay more for environment-friendly goods and services, how frequently they use home appliances in eco mode, and whether they use a reusable water bottle. Crucially, we also make use of respondent's reported votes in the municipal election of 2016, with the intention of absorbing to a significant extent the unobserved political preferences of respondents prior to the implementation of the car ban.

2.2 General assumptions

Our methods generally rely on assumptions of overlap and unconfoundedness. We discuss overlap in the AIPW section below, but it appears that generally speaking the probability of owning a banned car conditional on our observed covariates is bounded away from 0 and 1. In our interpretation, this suggests that we do have overlap in the sample. To justify unconfoundedness, we rely most heavily on the inclusion of previous voting patterns as a covariate. This suggests that both treated and untreated units would have similar baseline voting intentions in the absence of treatment. There is also the potential issue that owners of older diesel cars (those affected by the car ban) might be less environmentally conscious, and therefore respond differently to treatment than the control units would in a counterfactual world. We attempt to mitigate this by including survey data about environmental attitudes. Between these two kinds of covariates, we hope to reasonably justify unconfoundedness.

2.3 A Consideration: Inclusion of 65 Treated Respondents with Missing Covariates

65 of the respondents provided all covariate data except for the identifying information for their cars: eg., if it is a Euro4 model with Petrol, a Euro5 model with Diesel, etc. They self-reported if their car was banned or not (assignment into treatment or control). The paper estimates all the specifications both with and without these 65 respondents.

However, we decide to include them in our prediction and analysis because we do not use the car covariates when comparing our ML-based methods. The car covariates (general model version, and petrol v. diesel) determine exactly if the respondent is in treated or control group—if their car is banned or not. The treatment variable diesel_euro4_ass lets us include these 65 respondents in our dataset.

3 ATE

3.1 AIPW

We first attempted to estimate the Average Treatment Effect (ATE) using the AIPW method. This technique is a double robust estimator that utilizes both the propensity score model and the outcome model. For our implementation, we used forest-based estimates were used for both the outcome model and the propensity scores (causal forest with 100 trees). Our treatment variable was a dummy for owning a car that was banned under the Diesel Euro 4 standard. We hypothesized its potential influence on the chosen outcome variable: voting for Lega Euro.

In the causal forest model, we included all previously selected covariates, hoping these would control for confounding effects, allowing us to more accurately isolate the impact of our treatment variable on the outcome of interest. The AIPW method provided an ATE estimate of 0.0438, with a standard error of 0.0226. Notably, our estimate was less than half of the magnitude of that of the diff-in-diff estimate found in the original paper, although it maintained the same directional influence. We believed this to be a more realistic estimate of the treatment effect, in contrast to the more substantial effect identified in the paper. The inflated results reported in the paper could be a consequence of unobserved characteristics or inadequate control measures, which might have overstated the estimated treatment effect.

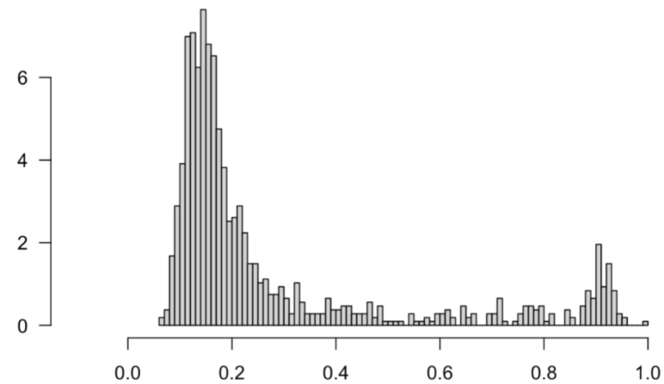


Figure 2: Estimated Propensity Scores

To further investigate our model, we plotted the propensity scores to evaluate their level of overlap. These scores appeared slightly clustered around 0.15, a reasonable outcome considering the rarity of owning any particular type of car. However, this propensity score distribution could have affected our AIPW estimator's performance. Limited overlap suggests that there were few individuals with similar probabilities of falling into either treatment or control groups. This potential lack of common support might have impacted the robustness of our estimates.

This is why we considered another method for estimating the ATE.

3.2 Parametric g-formula

An alternative to IPW is to use standardization. That is, instead of learning the treatment function, we construct a model to learn the mean conditional outcome function directly. In doing so, the parametric g-formula relies on the same assumptions as IPW, but in practice can perform better in settings with limited overlap.

The idea is simple. If we knew the true conditional mean outcome function, $E[Y|W, X]$, we could compute the true ATE by taking $E[Y|W = 1, X] - E[Y|W = 0, X]$. Since we do not know the function, we estimate it on the data using a parametric model, then estimate the ATE by computing counterfactual estimated outcomes for both the treatment and control groups. We use a logistic regression of the form

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 W + \beta_2 X)}},$$

since our outcome variable of interest is binary. After estimating this model to yield an estimated conditional mean outcome function $p(W, X)$, we compute our ATE estimate as

$$\frac{1}{n}\sum_{i}\left(p(1, X_i) - p(0, X_i)\right).$$

That is, we treat p as the true conditional mean outcome function, and compute the average difference between estimated treated and untreated outcomes for all observations in the sample (treated and untreated). We then use the delta method to estimate a standard error on our estimate for the construction of confidence intervals. Our results are below.

Table 1: Parametric g-formula results

ATE	SE	z	Pr(>|z|)
0.0798	0.0292	2.73	0.00625

Here, we obtain an ATE estimate greater than our AIPW estimate, but still below that of the original paper (0.135). This approach is limited by the fact that we make structural assumptions on the shape of the conditional mean outcome function in order to estimate it. We likely introduced bias if the true function is not logistic, or if there are unobserved variables that otherwise distinguish the treatment and control groups (in a way that violates unconfoundedness). This is at least somewhat likely, since we deal with observational data rather than an RCT.

4 HTE

4.1 R learner

The R-learner first estimates marginal effects and treatment propensities in order to form an objective function that isolates the causal component of the signal. Next, we optimize this data-adaptive objective function. For both steps, we can use any loss-minimization method, e.g., the lasso, random forests, boosting, etc.; moreover, these methods can be fine-tuned by cross validation. The particular form of the R-learner in our project is a lightweight implementation of the R-learner using the lasso.

We split our sample into 70 percent training data and 30 percent test data, for the purpose of training the lasso fit on the training set and making predictions on the test set. After attaining the R-learner CATE estimates, we test whether there is any notable heterogeneity present. The graph below shows the average AIPW scores within quintiles, as defined by the predicted CATEs. The plot is non-monotonic, suggesting that there is not much detectable heterogeneity. This can mean that the number of observations in our dataset is too small for us to find subgroups with relevant differences in treatment effect. Nevertheless, the jump of AIPW score at the fifth quintile is evidence for heterogeneity between the fifth quintile—predicted to have the largest treatment effect by the car ban—and the rest of the sample.

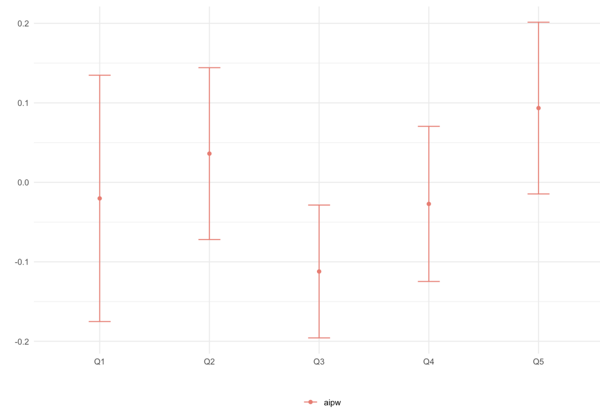


Figure 3: AIPW scores by CATE Estimate Ranking

Parallel to the above calculation of average AIPW score by group membership, Rank-Weighted Average Treatment Effect (RATE) is used to evaluate the R-learner predictions in distinguishing subpopulations with different treatment effects. Along the X-axis, we divide the population into groups defined by a certain prioritization rule, in this case, the CATE predictions from the R-learner. The Y-axis gives the differences between the ATE from treating up to q of the population and overall ATE from treating everyone as estimated by fitting a causal forest to the training set. We also attain an estimate, 0.065, for the area under the TOC.

Table 2: RATE with R-learner

estimate	std.err	target
0.0529	0.024	AUTOC

Despite the high noise-to-signal ratio, the lump at the start of the TOC curve is consistent with the jump of AIPW score from the previous figure, both of which would signify strongest treatment effect received by the top quintile in the sample, relative to the other four quintiles. We are, however, cautious in interpreting this as a definite claim, due to the wide confidence interval of our estimate.

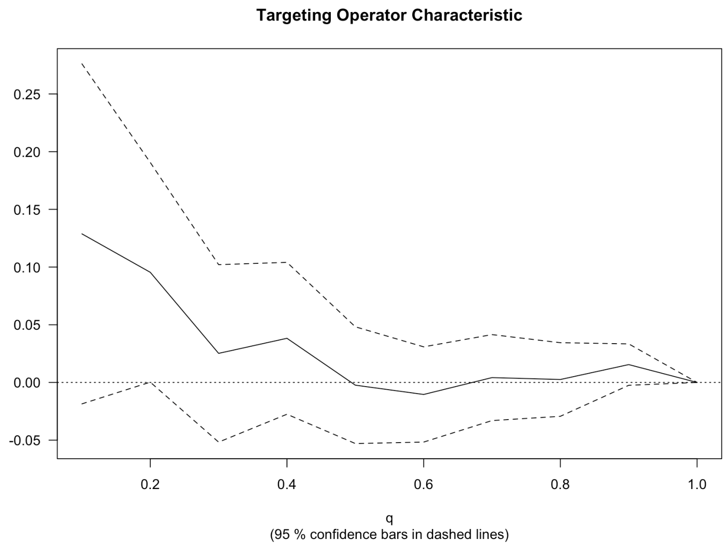


Figure 4: Targeting Operator Characteristic Using R-learner Predictions

Finally, to further assess the relationship between covariate and CATE, we build a heatmap of average covariate values within groups, similarly ranked by the CATE estimate. This is helpful for revealing any monotonic increase or decrease across rankings as well as describing the subgroups with strongest and weakest estimated treatment effect. The few indicators of people's attitudes towards environmental issues at the bottom of the table show average covariate values skewed towards individuals with higher estimated CATEs. Contrary to expectation, those estimated to be most likely voting for the conservative party after the car ban treatment are in fact not more hostile to environmental issues. If anything, they appear to display a higher degree of interest in environmental action, as seen in figure 5.

4.2 Causal Forest Approach

Next, we analyze HTE through the construction of a causal forest (for both training and test sets as a comparison). We want to get unbiased estimates of the CATE so we train our causal forest on our train data set, then use the test set to construct another causal forest. This second causal forest from the test set is used with a calibration test and then RATE to evaluate if there is any HTE.

We run a calibration test that estimates the best linear predictor of true CATE with our out-of-bag predictions, from our test data set. The results of the calibration test as are follows:

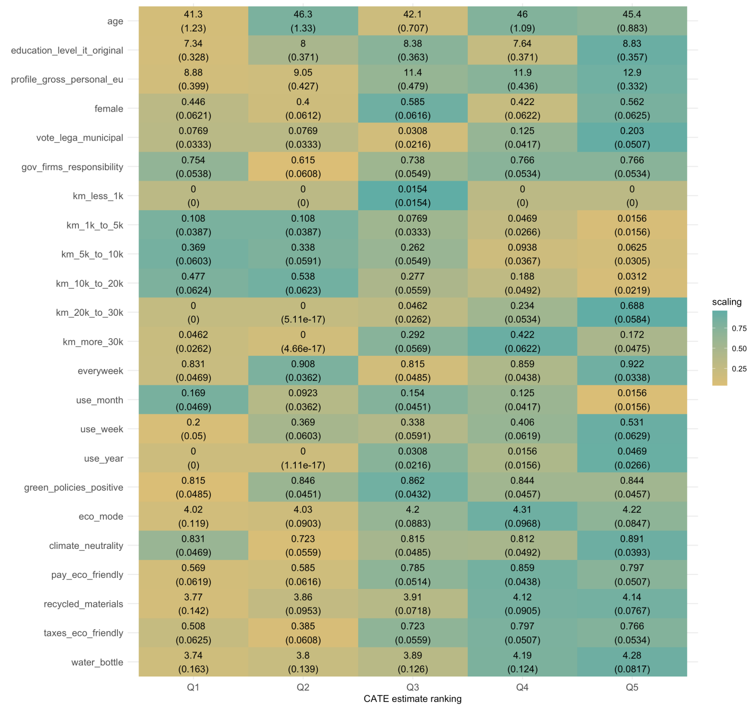


Figure 5: Heatmap of Average Covariate Values by CATE Estimate Ranking

Table 3: Forest Predictions and Heteroskedasticity-Robust SEs

	Estimate	Std. Error	t value	Pr(>t)	
mean.forest.prediction	-3.0206	4.2708	-0.7073	0.76004	
differential.forest.prediction	2.6976	1.5662	1.7224	0.04298	$*$

Note: Best linear fit using forest predictions (on held-out data) as well as the mean forest prediction as regressors, along with one-sided heteroskedasticity-robust (HC3) SEs.
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.

We initially and tentatively reject the null hypothesis of no HTE based on the differential.forest.prediction estimate being significant and above 0. However upon closer inspection, our mean.forest.prediction estimate is not close to 1 (being close to 1 would mean that the forest produces correct predictions). Instead, we get a negative estimate with large standard error. As such, we cannot trust that our calibration test is valid. And while the estimate mean.forest.prediction at a 95% CI covers 1, we really need a larger data set to confirm if the estimate is at 1 and to then rely on our significant finding of heterogeneity (rejection of non-heterogeneity).

Next, we use RATE to evaluate the test causal forest prediction, dividing the population on the X-axis, in ranked order, on the prioritization rule of CATE prediction as produced by our training set causal forest.

Table 4: RATE with another Causal Forest

estimate	std.err	target
0.1458	0.0298	AUTOC

We get an estimate for the area under the TOC curve of 0.1458, which is above 0 with very small standard errors (± 2 SEs still above 0), indicating HTE.

Next, we graph the TOC using the RATE function. As previously mentioned, the y-axis is the difference in ATE from that point's top-q proportion of the population versus the overall ATE from everyone being treated. So we expect it to decrease. Here that means that people most affected by the car ban based on their covariates are more likely to get a positive treatment effect: that is, they vote for the conservative party Lega.

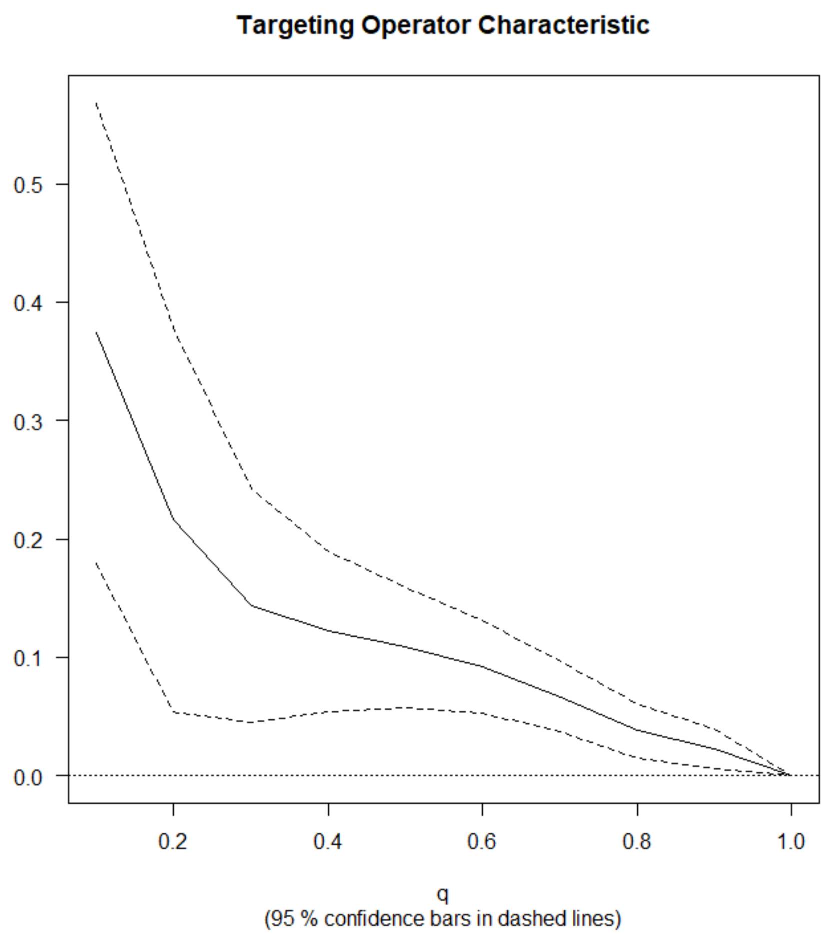


Figure 6: Targeting Operator Characteristic Using Causal Forest Predictions

As we can see, there is a noticeable decrease going from the most impacted by treatment to the rest of the population in their treatment effect compared to the ATE. This further supports an indication that there is HTE.

4.3 Best Linear Projection

Finally, for a richer understanding of the underlying general trends, we analyze our training set causal forest with the best_linear_projection() function, where we fit the conditional average treatment effect as a linear function of some intercept and the covariates (or a subset of the covariates, as we have here). Since we do not believe our model is truly linear in covariates, we take this as an exploratory approach only to suggest general trends, and not to imply causality from any of the covariates. We do not include the last 8 covariates due to table length, but their best linear projections were not significant. This full table is available on the next page as figure 5.

5 Concluding Remarks

Overall, we obtain an ATE estimate of 0.0438 from our AIPW model and of 0.0798 from our parametric g-forumla model. We find these smaller effect sizes to be more plausible that that found by the original paper.

For HTE, using the R-learner, we do not find notable heterogeneity across the sample. Nevertheless, our positive RATE estimate, along with a noticeable difference in AIPW scores between the group with the highest predicted CATE estimates and the rest of sample, provides indication HTE might exist.

Through the causal forest approach, we do find evidence against the null of no heterogeneity, with a TOC curve that stays above 0 for 80 percent of the population with narrow confidence intervals.

Table 5: Best Linear Projection of the CATE

	Estimate	Std. Error	t value	Pr(>t)	
(Intercept)	−0.0166969	0.5032418	−0.0332	0.97354	
age	0.0035305	0.0021488	1.6430	0.10082	
factor.eco_mode.2	−0.1226695	0.2662186	−0.4608	0.64510	
factor.eco_mode.3	0.0562728	0.2614194	0.2153	0.82963	
factor.eco_mode.4	−0.0269397	0.2562903	−0.1051	0.91632	
factor.eco_mode.5	−0.0929013	0.2557007	−0.3633	0.71647	
vote_lega_municipal	0.1405839	0.0617632	2.2762	0.02313	*
EDU1	0.0851354	0.2898156	0.2938	0.76903	
EDU2	0.1390340	0.3011904	0.4616	0.64450	
EDU3	0.0633945	0.2883324	0.2199	0.82604	
INC1	0.0623417	0.2840065	0.2195	0.82632	
INC2	0.1001200	0.1203893	0.8316	0.40590	
INC3	−0.1053028	0.1396244	−0.7542	0.45099	
INC4	−0.1485964	0.1472239	−1.0093	0.31317	
INC5	−0.0326339	0.1369418	−0.2383	0.81171	
INC6	−0.2129097	0.1335414	−1.5943	0.11131	
INC7	0.0047827	0.1589315	0.0301	0.97600	
INC8	0.0250331	0.1602542	0.1562	0.87591	
INC9	0.0295385	0.2147419	0.1376	0.89045	
INC10	0.0630143	0.1513456	0.4167	0.67759	
INC11	−0.1275691	0.1384262	−0.9216	0.35707	
INC12	−0.0343571	0.1805407	−0.1903	0.84913	
INC13	−0.2594215	0.1528363	−1.6974	0.09007	.
INC14	−0.0012047	0.1794959	−0.0067	0.99465	
INC15	−0.2147770	0.2171194	−0.9892	0.32290	
everyweek	0.1270643	0.2066526	0.6149	0.53884	
female	−0.0464232	0.0515831	−0.9000	0.36844	
gov_firms_responsibility	−0.0202847	0.0510915	−0.3970	0.69147	
green_policies_positive	0.0093778	0.0828160	0.1132	0.90988	
climate_neutrality	0.1107217	0.1169043	0.9471	0.34391	
km_1k_to_5k	−0.3225742	0.1829761	−1.7629	0.07835	.
km_5k_to_10k	−0.3093160	0.1898965	−1.6289	0.10379	
km_10k_to_20k	−0.2671178	0.1918483	−1.3923	0.16426	
km_20k_to_30k	−0.2479293	0.1981724	−1.2511	0.21132	
km_less_1k	−0.2155154	0.1977256	−1.0900	0.27610	
km_more_30k	−0.2188747	0.2141903	−1.0219	0.30719	
pay_eco_friendly	−0.0487920	0.0936485	−0.5210	0.60252	
factor.recycled_materials.2	−0.4738608	0.2229012	−2.1259	0.03386	*

Note: Confidence intervals are cluster- and heteroskedasticity-robust (HC3).

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1.

References

[1] Italo Colantone, Livio Di Lonardo, Yotam Margalit, and Marco Percoco. Replication data for: The political consequences of green policies: Evidence from italy, 2023.

[2] Susan Athey, Guido Imbens, Yanyang Kong, and Vikas Ramachandra. An introduction to recursive partitioning for heterogeneous causal effects estimation using causaltree package. *Github*, 2016.

A Appendix: Exploratory Work of HTE Through Honest Causal Trees

A.1 Honest Causal Trees Description

We use honest causal trees, which build binary regression tree models in two stages in order to estimate heterogeneous treatment effects. We have a binary treatment variable (alongside a binary outcome variable), so we can use the causal tree method, though we do not have a randomized setting and as such this method is invalid for real results and only used in the context of this appendix, for exploratory work.

We first build an unpruned honest causal tree, then a pruned one using cross-validation to find the optimal complexity parameter. We must note caution in the interpretation of the results of the splits generated from the trees, as a split on a variable could simply mean it is correlated with another variable that is explanatory of underlying effects.

A.2 Further Data Preparation

Honest causal trees required us to redefine two covariates from our data preparation file ("factor(eco_mode)" and "factor(recycled_materials)") back to their original factor form and not as dummy indicator variables.

Then, as described in the documentation to the causalTree package, "honesty" of our causal tree signifies that we "estimate causal effects in the leaves of a given tree on an independent estimation sample rather than the data used to build and cross-validate the tree" [2].

As such, we split our training data set (which compromises 70 percent of the total data) into two smaller data set, one to build and prune the tree, called split data, and the other to estimate the causal effects, called est data. We make careful note to balance these data set with adequate number of treated and control, so we split first on treatment and control and then merge into "split_data" and "est_data".

A.3 Results

Our unpruned tree splits on a low income bracket (INC5), then further on gender as the first two splits.

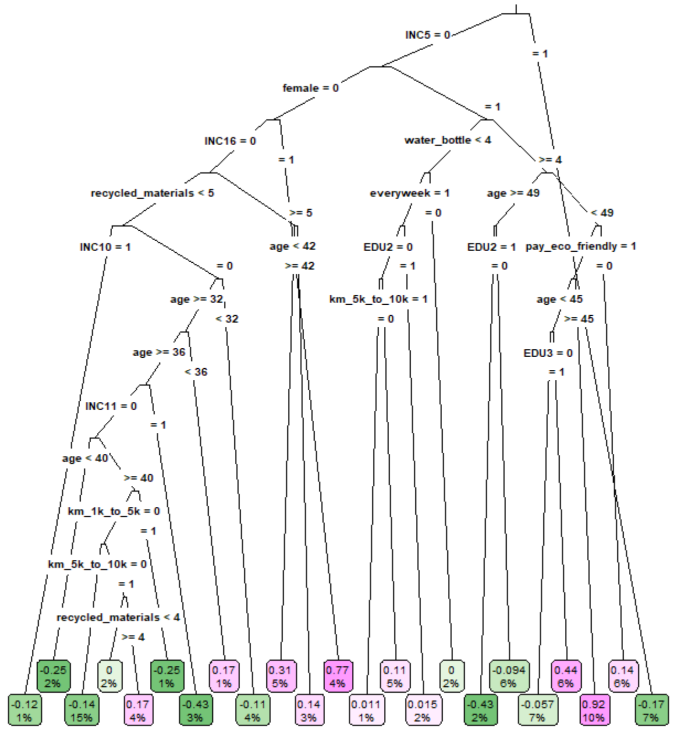


Figure 7: Unpruned Honest Causal Tree for HTE

However when we use cross-validation to produce the optimal parameters, and then prune our tree, we get the following:

Figure 8: Pruned Honest Causal Tree for HTE

The tree does not split, implying no heterogeneity in the treatment effect, though as noting above we must not take this result too seriously.

Data Preperation

Krista A., Andrew C., Patricio O., Diane S.

2023-06-07

```
library(haven)
data <- read_dta("Replication_Dataset.dta")

Y <- data$vote_lega_euro
W <- data$diesel_euro4_ass

covariates <- c("age", "factor(eco_mode)", "vote_lega_municipal", "EDU1", "EDU2", "EDU3",
                "INC1", "INC2", "INC3", "INC4", "INC5", "INC6", "INC7", "INC8", "INC9",
                ↪  "INC10", "INC11", "INC12",
                "INC13", "INC14", "INC15", "INC16","everyweek", "female",
                ↪  "gov_firms_responsibility",
                "green_policies_positive", "climate_neutrality", "km_1k_to_5k",
                ↪  "km_5k_to_10k",
                "km_10k_to_20k", "km_20k_to_30k", "km_less_1k", "km_more_30k",
                ↪  "pay_eco_friendly",
                "factor(recycled_materials)", "taxes_eco_friendly", "use_month",
                ↪  "use_week", "use_year",
                "water_bottle")

X <- model.matrix(formula(paste0("~", paste0(covariates, collapse="+"))), data=data)
X <- X[, -1]


#Note that INC15 is a dummy if income question does not apply, and INC16 is prefer not to
↪  respond

# train-test split
# Separate treatment and control groups
treatment_data <- data[data$diesel_euro4_ass == 1, ]
control_data <- data[data$diesel_euro4_ass == 0, ]

# Set seed for reproducibility
set.seed(42)

# Split treatment group into train and test datasets
treatment_train <- treatment_data[sample(nrow(treatment_data), floor(0.7 *
↪  nrow(treatment_data))), ]
treatment_test <- treatment_data[!(rownames(treatment_data) %in%
↪  rownames(treatment_train)), ]

# Split control group into train and test datasets
control_train <- control_data[sample(nrow(control_data), floor(0.7 *
↪  nrow(control_data))), ]
```

```r
control_test <- control_data[!(rownames(control_data) %in% rownames(control_train)), ]

# Merge train datasets of treatment and control groups
train_data <- rbind(treatment_train, control_train)

# Merge test datasets of treatment and control groups
test_data <- rbind(treatment_test, control_test)

Y_train <- train_data$vote_lega_euro
W_train <- train_data$diesel_euro4_ass
X_train <- model.matrix(formula(paste0("~", paste0(covariates, collapse="+"))),
↪  data=train_data)
X_train <- X_train[, -1]

Y_test <- test_data$vote_lega_euro
W_test <- test_data$diesel_euro4_ass
X_test <- model.matrix(formula(paste0("~", paste0(covariates, collapse="+"))),
↪  data=test_data)
X_test <- X_test[, -1]
```

AIPW for project

Diane Sarkis

06/07/2023

```r
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(sandwich)
library(grf)
library(glmnet)
```

```
## Loading required package: Matrix

## Loaded glmnet 4.1-7
```

```r
library(splines)
library(ggplot2)
library(reshape2)
library(RColorBrewer)
```

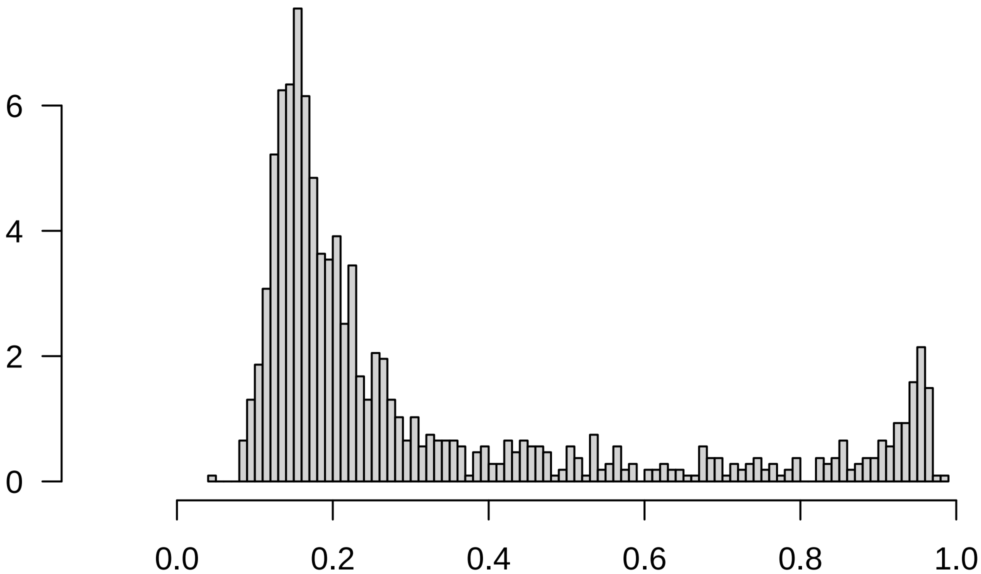
```r
# Input covariates need to be numeric.

# Estimate a causal forest.
forest <- causal_forest(X=X,W=as.vector(W), Y=as.vector(Y), num.trees = 100)
forest.ate <- average_treatment_effect(forest)
```

```
## Warning in average_treatment_effect(forest): Estimated treatment propensities
## take values between 0.047 and 0.985 and in particular get very close to 0 and
## 1. In this case, using `target.sample=overlap`, or filtering data as in Crump,
## Hotz, Imbens, and Mitnik (Biometrika, 2009) may be helpful.
```

```r
print(forest.ate)
```

```
##   estimate    std.err
## 0.03765487 0.02196327
```

```r
e.hat <- forest$W.hat
hist(e.hat, main="", breaks=100, freq=FALSE,
     xlab="", ylab="", xlim=c(-.1, 1.1), las=1)
```



PAST CODE # {r} # W_formula <- as.formula(paste("W ~", paste(covariates, collapse="+"))) # # Fit the logistic regression model # W_model <- glm(W_formula, family=binomial(link="logit"), data=data) # # Obtain predicted probabilities of treatment # W_hat <- predict(W_model, type="response") # # Define the outcome model # Y_formula <- as.formula(paste("Y ~ W +", paste(covariates, collapse="+"))) # # Fit the linear regression model # Y_model <- glm(Y_formula, data=data) # # Obtain predicted outcomes # Y_hat <- predict(Y_model) # # Compute weights for AIPW # weights <- ifelse(W==1, 1/W_hat, 1/(1-W_hat)) # # # Compute AIPW estimate of ATE # ATE_AIPW <- mean((W - W_hat)*(Y - Y_hat)*weights + W_hat*Y_hat - (1-W_hat)*(1-Y_hat)*weights) #{r} # ATE_AIPW “‘ Fit two models: a logistic regression to estimate the probability of treatment, and a linear regression to estimate the potential outcome under treatment and control. We then compute the AIPW estimate of the ATE.

ATE Estimation with G-learner

Andrew Conkey

2023-06-07

```r
library(haven)
library(boot)
library(marginaleffects)
```

```r
df <- read_dta("Replication_Dataset.dta")


f2 <- vote_lega_euro ~ diesel_euro4_ass + age + factor(eco_mode) + vote_lega_municipal +
↪  EDU1 + EDU2 + EDU3 +
  factor(profile_gross_personal_eu) + everyweek + female +
  gov_firms_responsibility + green_policies_positive + climate_neutrality +
  km_1k_to_5k + km_5k_to_10k + km_10k_to_20k + km_20k_to_30k +
  km_less_1k + km_more_30k + pay_eco_friendly +
  factor(recycled_materials) + taxes_eco_friendly +
  use_month + use_week + use_year + water_bottle + dummy_buy + dummy_donation +
↪  dummy_genitori_click + dummy_newsletter +
  dummy_podcast + dummy_social + dummy_watch_video + dummy_zeroco2_click

fit2 <- glm(f2, data = df, family=binomial)

avg_comparisons(fit2, variables = list(diesel_euro4_ass = 0:1))
```

```
##
##              Term Contrast Estimate Std. Error    z Pr(>|z|)  2.5 % 97.5 %
##  diesel_euro4_ass    1 - 0   0.0793     0.0308 2.57   0.0102 0.0188   0.14
##
## Columns: term, contrast, estimate, std.error, statistic, p.value, conf.low, conf.high
```

Using R Learner to Estimate HTE

Krista Arenaodu

2023-06-07

```
library(haven)
library(dplyr)
library(lmtest)
library(sandwich)
library(glmnet)
library(grf)
library(devtools)
```

```
data$education_level_it_original <- ifelse(data$education_level_it_original == 13 |
↪  data$education_level_it_original == 14, NA, data$education_level_it_original)
data$profile_gross_personal_eu <- ifelse(data$profile_gross_personal_eu == 98 |
↪  data$profile_gross_personal_eu == 99, NA, data$profile_gross_personal_eu)
```

Implementing R-learner

```
#install_github("xnie/rlearner", force=TRUE)
library(rlearner)

# train lasso fit on the training set
rlasso_fit = rlasso(X_train, W_train, Y_train)
# make predictions on the test set
rlasso_est = predict(rlasso_fit, X_test)

test_data$tau <- rlasso_est
test_data <- mutate(test_data, ranking = ntile(test_data$tau, 5)) # create tau estimate
↪  ranking
```

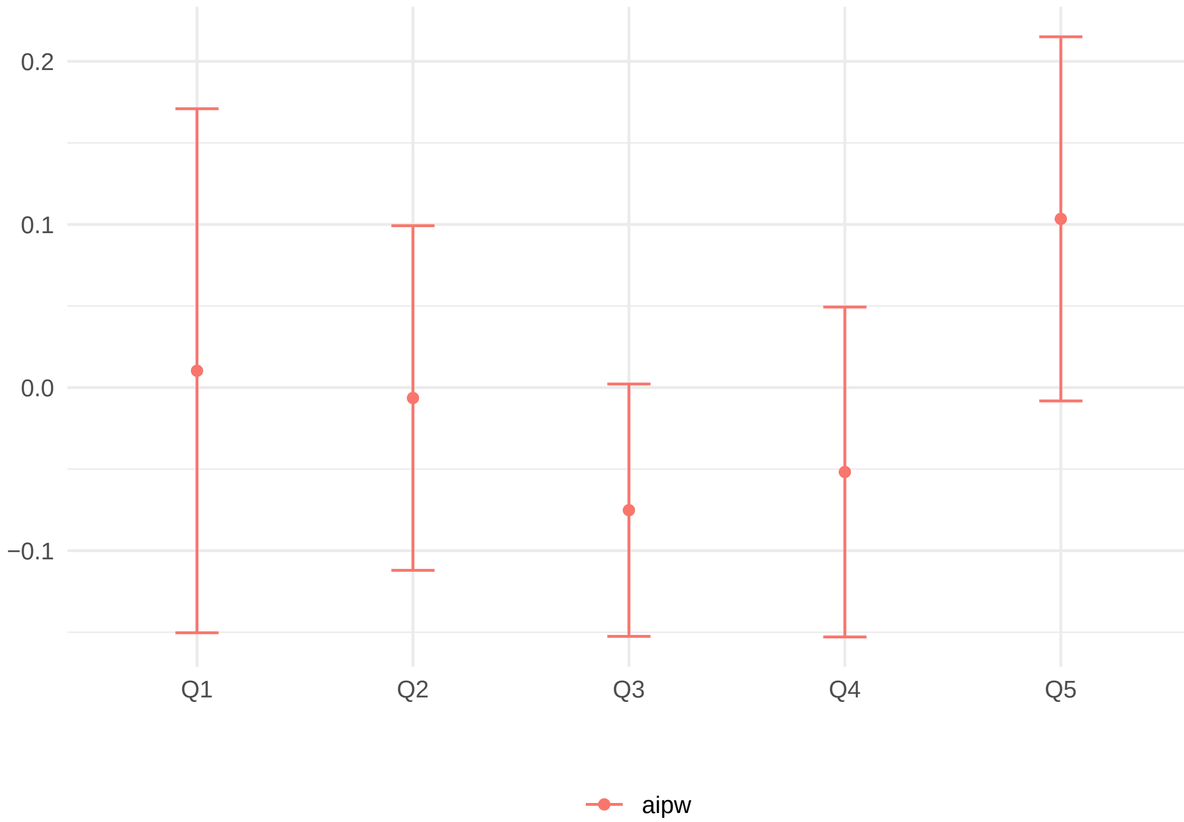
AIPW-CATE graph

```
# AIPW estimates on the Y-axis, quartiles as defined by the R learner on the X-axis

# estimating AIPW estimator using causal forest
# Observational setting with unconf + overlap, unknown assignment probs.
forest.tau <- causal_forest(X_test, as.vector(Y_test), as.vector(W_test))

# Get forest predictions.
tau.hat <- predict(forest.tau)$predictions
m.hat <- forest.tau$Y.hat  # E[Y|X] estimates
e.hat <- forest.tau$W.hat  # e(X) := E[W|X] estimates (or known quantity)
```

```r
tau.hat <- forest.tau$predictions  # tau(X) estimates

# Predicting mu.hat(X[i], 1) and mu.hat(X[i], 0) for obs in held-out sample
# Note: to understand this, read equations 6-8 in this vignette
# https://grf-labs.github.io/grf/articles/muhats.html
mu.hat.0 <- m.hat - e.hat * tau.hat        # E[Y|X,W=0] = E[Y|X] - e(X)*tau(X)
mu.hat.1 <- m.hat + (1 - e.hat) * tau.hat  # E[Y|X,W=1] = E[Y|X] + (1 - e(X))*tau(X)

# Compute AIPW scores
aipw.scores <- tau.hat + W_test / e.hat * (Y_test -  mu.hat.1) - (1 - W_test) / (1 -
↪  e.hat) * (Y_test -  mu.hat.0)
test_data$aipw.scores <- aipw.scores

# Estimate average treatment effect conditional on group membership
ols <- lm(aipw.scores ~ 0 + factor(ranking), test_data)
forest.ate <- data.frame("aipw", paste0("Q", seq(5)), coeftest(ols, vcov=vcovHC(ols,
↪  "HC2"))[,1:2])
colnames(forest.ate) <- c("method", "ranking", "estimate", "std.err")
rownames(forest.ate) <- NULL # just for display
forest.ate
```

```
##   method ranking     estimate    std.err
## 1   aipw      Q1  0.010299701 0.08031645
## 2   aipw      Q2 -0.006417972 0.05281480
## 3   aipw      Q3 -0.075183068 0.03868018
## 4   aipw      Q4 -0.051771804 0.05056031
## 5   aipw      Q5  0.103413431 0.05581470
```

```r
# Plotting the point estimate of average treatment effect
# and 95% confidence intervals around it.
library(ggplot2)
ggplot(forest.ate) +
  aes(x = ranking, y = estimate, group=method, color=method) +
  geom_point(position=position_dodge(0.2)) +
  geom_errorbar(aes(ymin=estimate-2*std.err, ymax=estimate+2*std.err), width=.2,
  ↪  position=position_dodge(0.2)) +
  ylab("") + xlab("") +
  theme_minimal() +
  theme(legend.position="bottom", legend.title = element_blank())
```



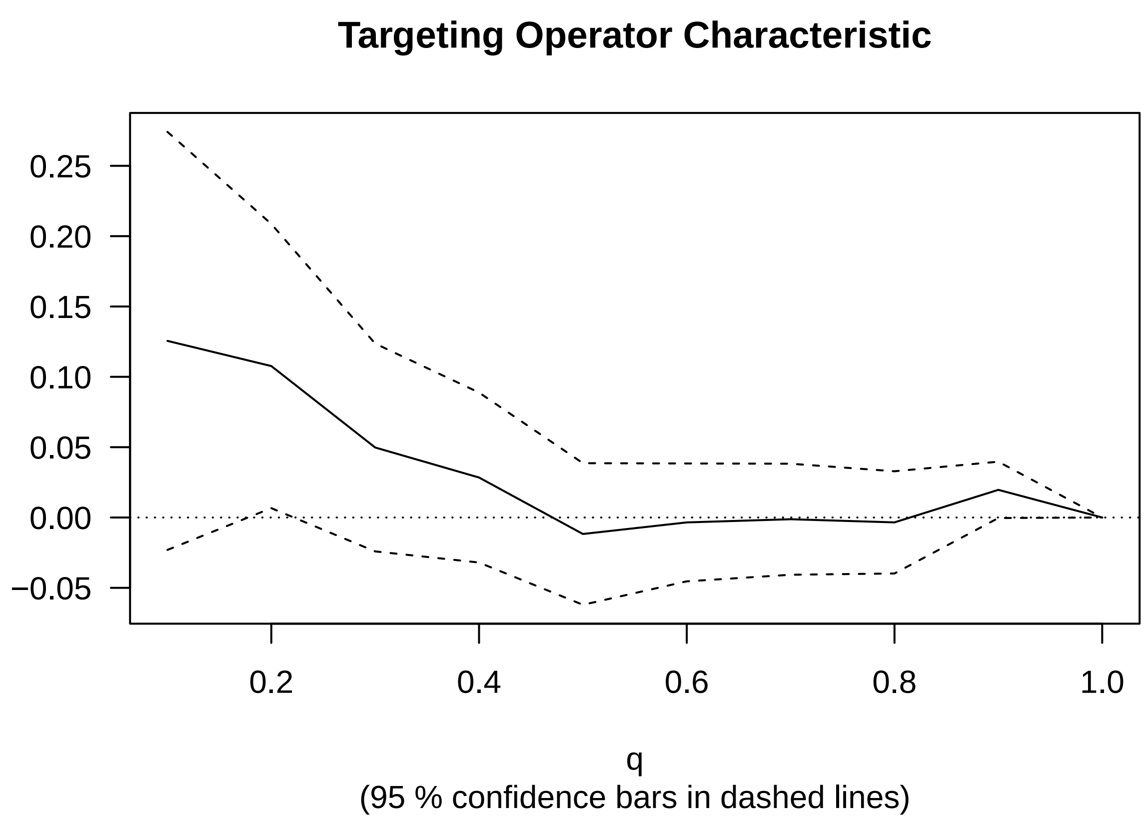
RATE curve

causal forest estimates on the Y-axis, prioritization based on R learner on the X-axis

```
rank_rlearner <- rank_average_treatment_effect(forest.tau, rlasso_est)
rank_rlearner
```

```
##    estimate    std.err     target
##  0.05816311 0.02448963 V1 | AUTOC
```

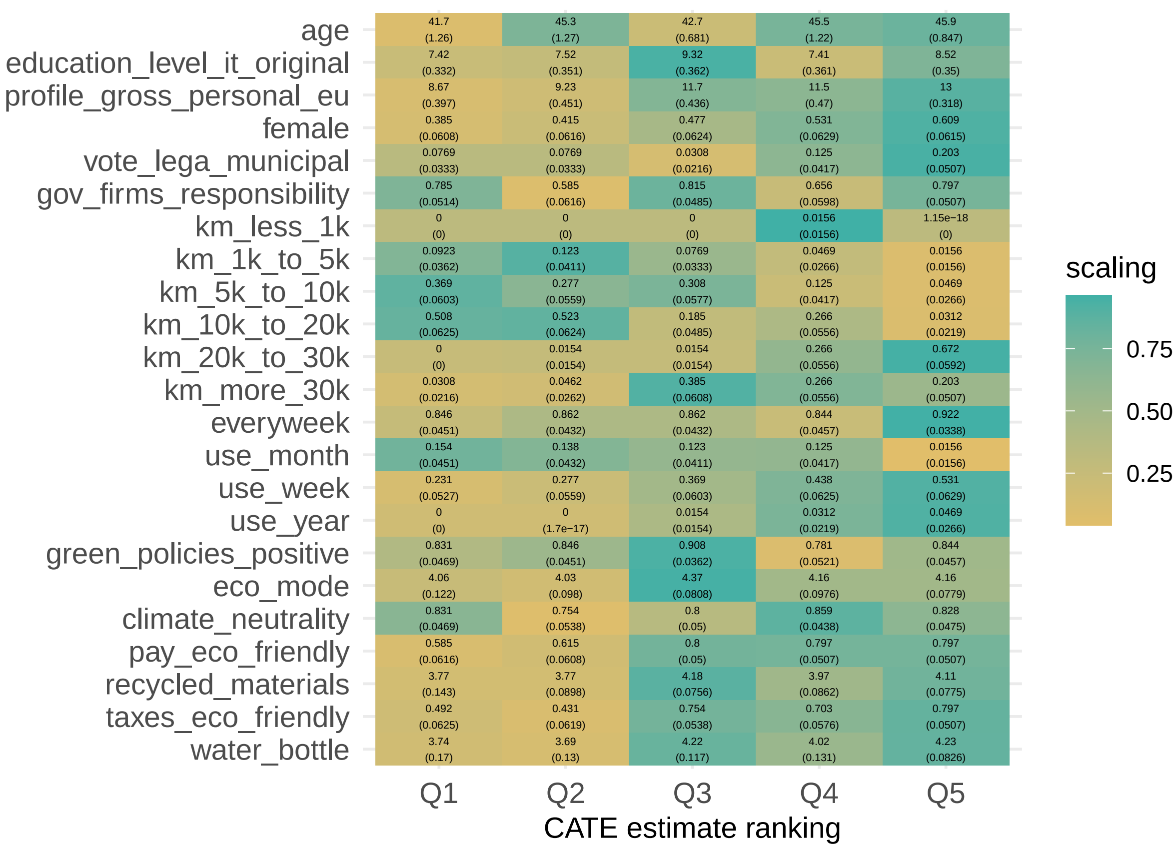
```
plot(rank_rlearner, las=1)
```



Plot heatmap

```
covariates_display <- c("age", "education_level_it_original",
↪  "profile_gross_personal_eu",
                        "female", "vote_lega_municipal", "gov_firms_responsibility",
                        "km_less_1k", "km_1k_to_5k", "km_5k_to_10k", "km_10k_to_20k",
                        ↪  "km_20k_to_30k",
                        "km_more_30k", "everyweek", "use_month", "use_week", "use_year",
                        ↪  "green_policies_positive",
                        "eco_mode", "climate_neutrality", "pay_eco_friendly",
                        ↪  "recycled_materials", "taxes_eco_friendly",
                        "water_bottle")

df <- mapply(function(covariates_display) {
  # Looping over covariate names
  # Compute average covariate value per ranking (with correct standard errors)
  fmla <- formula(paste0(covariates_display, "~ 0 + ranking"))
  ols <- lm(fmla, data=transform(test_data, ranking=factor(ranking)))
  ols.res <- coeftest(ols, vcov=vcovHC(ols, "HC2"))

  # Retrieve results
  avg <- ols.res[,1]
  stderr <- ols.res[,2]

  data.frame(covariates_display, avg, stderr, ranking=paste0("Q", seq(5)),
          # Used for coloring
```

```r
            scaling=pnorm((avg - mean(avg))/sd(avg)),
            # We will order based on how much variation is 'explained' by the averages
            # relative to the total variation of the covariate in the data
            variation=sd(avg) / sd(unlist(test_data[,covariates_display])),
            # String to print in each cell in heatmap below
            labels=paste0(signif(avg, 3), "\n", "(", signif(stderr, 3), ")"))
}, covariates_display, SIMPLIFY = FALSE)
df <- do.call(rbind, df)

# a small optional trick to ensure heatmap will be in decreasing order of 'variation'
# df$covariates_display <- reorder(df$covariates_display, order(df$variation))
# Convert the 'covariates_display' column to a factor with original order
df$covariates_display <- factor(df$covariates_display, levels =
↪  rev(unique(df$covariates_display)))

# plot heatmap
ggplot(df) +
  aes(ranking, covariates_display) +
  geom_tile(aes(fill = scaling), width = 1, height = 1) +
  geom_text(aes(label = labels), size = 1.5) +
  scale_fill_gradient(low = "#E1BE6A", high = "#40B0A6") +
  theme_minimal() +
  ylab("") + xlab("CATE estimate ranking") +
  theme(plot.title = element_text(size = 11, face = "bold"),
        axis.text=element_text(size = 11))
```



```
# What describes the subgroups with strongest and weakest estimated treatment effect?
# Are there variables that seem to increase or decrease monotonically across rankings?
```

HTE: Causal Forest

Patricio Ortiz

2023-06-07

Section 0: Import Data, Clean

```r
#Install Packages, uncomment first 2 lines
  #install.packages("devtools")  # if you don't have this installed yet.
  #devtools::install_github('susanathey/causalTree')

library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.2.3
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(plm)
```

```
## Warning: package 'plm' was built under R version 4.2.3
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```r
#Import Data from data preparation file, Note to future user: change to your working
↪  directory!
wd = getwd()
source(paste0(wd, "/data preparation.R"))
```

Section 1: Causal Forest

```r
# estimating AIPW estimator using causal forest
    library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.2.3

## Loading required package: Matrix

## Loaded glmnet 4.1-7
```

```r
    library(grf)
```

```
## Warning: package 'grf' was built under R version 4.2.3
```

```r
    # Observational setting with unconf + overlap, unknown assignment probs. Use test
    ↪  data set.
    train.forest.tau <- causal_forest(X_train, as.vector(Y_train), as.vector(W_train))
    test.forest.tau <- causal_forest(X_test, as.vector(Y_test), as.vector(W_test))
    #new  tau.hat.est
    tau.hat.est2 <- predict(train.forest.tau, X_test)$predictions
```

Section 2: Calibration Test

```r
test_calibration(test.forest.tau)
```

```
##
## Best linear fit using forest predictions (on held-out data)
## as well as the mean forest prediction as regressors, along
## with one-sided heteroskedasticity-robust (HC3) SEs:
##
##                                Estimate Std. Error t value  Pr(>t)
## mean.forest.prediction          -2.5512     3.0908 -0.8254 0.79513
## differential.forest.prediction   2.9817     1.5845  1.8818 0.03039 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
      #coefficient of 2.7 with significance above 0. Where if this beta coefficient
        #was >0 significantly, then we can reject the null hypothesis of NO
        ↪  heterogeneity.
        #seeing as we see this, we reject the null hypothesis of no heterogeneity in
        ↪  treatment effect.
```

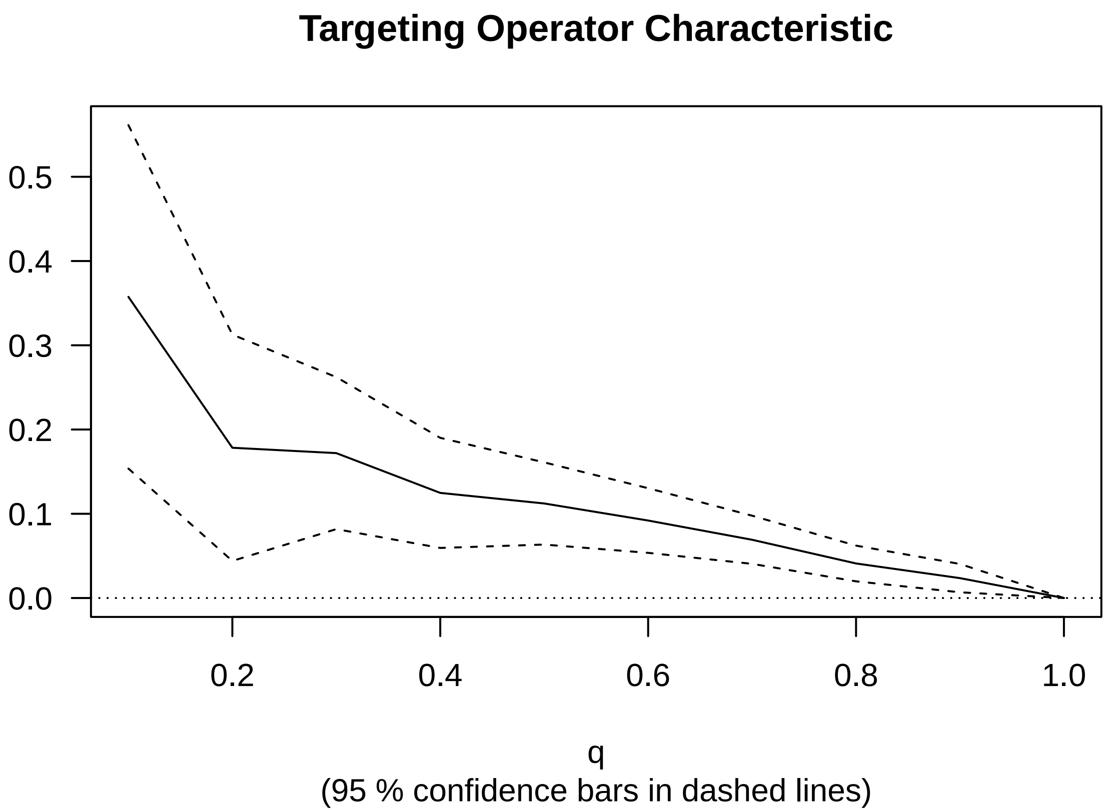
Section 3: Rate Curve:

y-axis: test causal forest predictions, x-axis: train data set causal forest predictions.

```r
rank_rlearner <- rank_average_treatment_effect(test.forest.tau, tau.hat.est2)
    rank_rlearner
```

```
##   estimate    std.err             target
##  0.1563209 0.02793894 priorities | AUTOC
```

```r
    plot(rank_rlearner, las=1)
```




```r
      #graph: estimate returns from RATE (on y-axis) represents the difference of the ATE
      ↪  from that point's top-q % of
        #population versus the overall ATE from everyone being treated. So we expect it
        ↪  to decrease.
        #Here that means that people most affected by the car ban based on their
        ↪  covariates are more likely to get
        #a positive treatment: that is, they vote for the conservative

      #estimate
        #positive and within +-2 standard errors is still above zero so heterogeneity is
        ↪  detected here at 95% CI.
```

Section 4: Best Linear Projections:

```
best_linear_projection(train.forest.tau, X_train)
```

```
##
## Best linear projection of the conditional average treatment effect.
## Confidence intervals are cluster- and heteroskedasticity-robust (HC3):
##
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -0.0166969  0.5032418 -0.0332  0.97354
## age                           0.0035305  0.0021488  1.6430  0.10082
## factor.eco_mode.2            -0.1226695  0.2662186 -0.4608  0.64510
## factor.eco_mode.3             0.0562728  0.2614194  0.2153  0.82963
## factor.eco_mode.4            -0.0269397  0.2562903 -0.1051  0.91632
## factor.eco_mode.5            -0.0929013  0.2557007 -0.3633  0.71647
## vote_lega_municipal           0.1405839  0.0617632  2.2762  0.02313 *
## EDU1                          0.0851354  0.2898156  0.2938  0.76903
## EDU2                          0.1390340  0.3011904  0.4616  0.64450
## EDU3                          0.0633945  0.2883324  0.2199  0.82604
## INC1                          0.0623417  0.2840065  0.2195  0.82632
## INC2                          0.1001200  0.1203893  0.8316  0.40590
## INC3                         -0.1053028  0.1396244 -0.7542  0.45099
## INC4                         -0.1485964  0.1472239 -1.0093  0.31317
## INC5                         -0.0326339  0.1369418 -0.2383  0.81171
## INC6                         -0.2129097  0.1335414 -1.5943  0.11131
## INC7                          0.0047827  0.1589315  0.0301  0.97600
## INC8                          0.0250331  0.1602542  0.1562  0.87591
## INC9                          0.0295385  0.2147419  0.1376  0.89063
## INC10                        -0.2162439  0.1554333 -1.3912  0.16459
## INC11                        -0.1275691  0.1384262 -0.9216  0.35707
## INC12                        -0.0343571  0.1805407 -0.1903  0.84913
## INC13                        -0.2594215  0.1528363 -1.6974  0.09007 .
## INC14                        -0.0012047  0.1794959 -0.0067  0.99465
## INC15                        -0.2147770  0.2171194 -0.9892  0.32290
## everyweek                     0.1270643  0.2066526  0.6149  0.53884
## female                       -0.0464232  0.0515831 -0.9000  0.36844
## gov_firms_responsibility     -0.0202847  0.0510915 -0.3970  0.69147
## green_policies_positive       0.0093778  0.0828160  0.1132  0.90988
## climate_neutrality            0.1107217  0.1169043  0.9471  0.34391
## km_1k_to_5k                  -0.3225742  0.1829761 -1.7629  0.07835 .
## km_5k_to_10k                 -0.3093160  0.1898965 -1.6289  0.10379
## km_10k_to_20k                -0.2671178  0.1918483 -1.3923  0.16426
## km_20k_to_30k                -0.2479293  0.1981724 -1.2511  0.21132
## km_less_1k                   -0.2155154  0.1977256 -1.0900  0.27610
## km_more_30k                  -0.2188747  0.2141903 -1.0219  0.30719
## pay_eco_friendly             -0.0487920  0.0936485 -0.5210  0.60252
## factor.recycled_materials.2 -0.4738608  0.2229012 -2.1259  0.03386 *
## factor.recycled_materials.3 -0.2291360  0.1666102 -1.3753  0.16948
## factor.recycled_materials.4 -0.1974741  0.1741960 -1.1336  0.25734
## factor.recycled_materials.5 -0.1034303  0.1847669 -0.5598  0.57580
## taxes_eco_friendly           -0.0047039  0.0931672 -0.0505  0.95975
## use_month                     0.1579871  0.2102435  0.7514  0.45263
```

```
## use_week                    0.0859162  0.0543385  1.5811  0.11430
## use_year                    0.2886135  0.2506657  1.1514  0.24996
## water_bottle                0.0343707  0.0213953  1.6065  0.10862
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Honest Causal Tree

Patricio Ortiz

2023-06-07

Section 0.A: Import Data, Clean

Here we import the data and redo the data preparation file for the 2 covariates in their full factor form and not as dummy indicator variables.

```
#Install Packages, uncomment first 2 lines
  #install.packages("devtools")  # if you don't have this installed yet.
  #devtools::install_github('susanathey/causalTree')
library(haven)
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.2.3

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(causalTree)
```

```
## Loading required package: rpart

## Warning: package 'rpart' was built under R version 4.2.3

## Loading required package: rpart.plot

## Warning: package 'rpart.plot' was built under R version 4.2.3

## Loading required package: data.table

## Warning: package 'data.table' was built under R version 4.2.3
```

```r
library(plm)
```

```
## Warning: package 'plm' was built under R version 4.2.3

##
## Attaching package: 'plm'

## The following object is masked from 'package:data.table':
##
##     between
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```r
#Would import data from data preparation file,
  #but need to change covariates object to covariates_honest in all instances
  #REDO  data preparation.R on lines 30-61
data <- read_dta("Replication_Dataset.dta")

Y <- data$vote_lega_euro
W <- data$diesel_euro4_ass

covariates_honest <- c("age", "eco_mode", "vote_lega_municipal", "EDU1", "EDU2", "EDU3",
                "INC1", "INC2", "INC3", "INC4", "INC5", "INC6", "INC7", "INC8", "INC9",
                ↪  "INC10", "INC11", "INC12",
                "INC13", "INC14", "INC15", "INC16","everyweek", "female",
                ↪  "gov_firms_responsibility",
                "green_policies_positive", "climate_neutrality", "km_1k_to_5k",
                ↪  "km_5k_to_10k",
                "km_10k_to_20k", "km_20k_to_30k", "km_less_1k", "km_more_30k",
                ↪  "pay_eco_friendly",
                "recycled_materials", "taxes_eco_friendly", "use_month", "use_week",
                ↪  "use_year",
                "water_bottle")

X <- model.matrix(formula(paste0("~", paste0(covariates_honest, collapse="+"))),
↪  data=data)
X <- X[, -1]

# train-test split
# Separate treatment and control groups
treatment_data <- data[data$diesel_euro4_ass == 1, ]
control_data <- data[data$diesel_euro4_ass == 0, ]
# Set seed for reproducibility
set.seed(42)

# Split treatment group into train and test datasets
treatment_train <- treatment_data[sample(nrow(treatment_data), floor(0.7 *
↪  nrow(treatment_data))), ]
treatment_test <- treatment_data[!(rownames(treatment_data) %in%
↪  rownames(treatment_train)), ]
```

```r
# Split control group into train and test datasets
control_train <- control_data[sample(nrow(control_data), floor(0.7 *
↪  nrow(control_data))), ]
control_test <- control_data[!(rownames(control_data) %in% rownames(control_train)), ]

# Merge train datasets of treatment and control groups
train_data <- rbind(treatment_train, control_train)

# Merge test datasets of treatment and control groups
test_data <- rbind(treatment_test, control_test)

Y_train <- train_data$vote_lega_euro
W_train <- train_data$diesel_euro4_ass
X_train <- model.matrix(formula(paste0("~", paste0(covariates_honest, collapse="+"))),
↪  data=train_data)
X_train <- X_train[, -1]

Y_test <- test_data$vote_lega_euro
W_test <- test_data$diesel_euro4_ass
X_test <- model.matrix(formula(paste0("~", paste0(covariates_honest, collapse="+"))),
↪  data=test_data)
X_test <- X_test[, -1]
```

Section 0.B: More Data Preperation: Further Train-Train Split

As noted in the write-up, we further split hte 70% training data by half into a train-train data set and an train-estimate data set to prune our causal trees of interest.

See output for full formula of outcome against full covariates.

```r
#Fix covariate list since looping over it, remove dummy variables:
extra_covariates_factor <- c("eco_mode", "recycled_materials", "age",
↪  "vote_lega_municipal", "EDU1")

train_data = as.data.frame(cbind(X_train,W_train,Y_train))
test_data = as.data.frame(cbind(X_test,W_test,Y_test))


fmla <- paste("Y_train", "~", paste(covariates_honest, collapse = " + "))
head(fmla)
```

```
## [1] "Y_train ~ age + eco_mode + vote_lega_municipal + EDU1 + EDU2 + EDU3 + INC1 + INC2 + INC3 + INC4
```

```r
# 2 subset split of training into split and estimates for creating causal tree and then
↪  checking it:
indices = split(seq(nrow(train_data)), sort(seq(nrow(train_data)) %% 2))
names(indices) = c('split', 'est')

#split into 50/50 split and est from training, both with certain number of control and
↪  treated samples
new_treatment_data <- train_data[train_data$W_train == 1, ]
```

```
new_control_data <- train_data[train_data$W_train== 0, ]

treatment_split <- new_treatment_data[sample(nrow(new_treatment_data), floor(0.5 *
↪  nrow(new_treatment_data))), ]
treatment_est <- new_treatment_data[!(rownames(new_treatment_data) %in%
↪  rownames(treatment_split)), ]
control_split <- new_control_data[sample(nrow(new_control_data), floor(0.5 *
↪  nrow(new_control_data))), ]
control_est <- new_control_data[!(rownames(new_control_data) %in%
↪  rownames(control_split)), ]

# Merge train datasets of treatment and control groups
split_data <- rbind(treatment_split, control_split)

# Merge test datasets of treatment and control groups
est_data <- rbind(treatment_est, control_est)
```

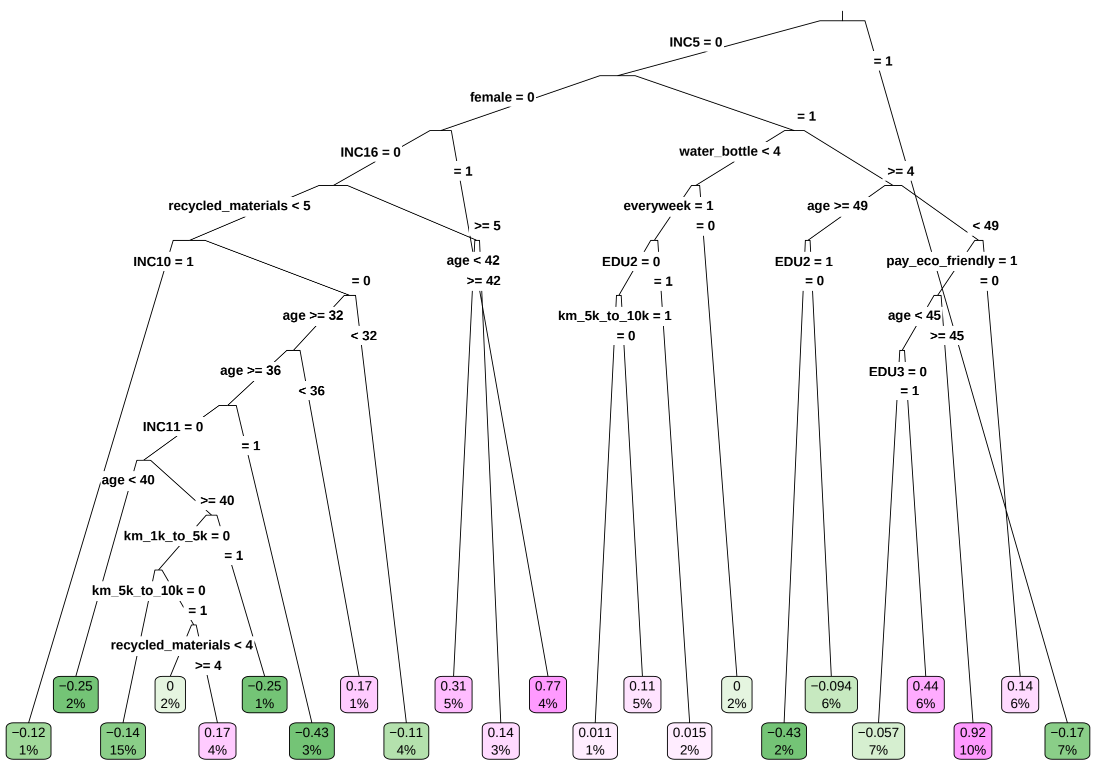
Section 1: Fit Causal Tree, Unpruned, Pruned

```
# Fit tree
ct.unpruned <- honest.causalTree(
  formula=fmla,             # Define the model
  data=split_data,
  treatment=split_data$W_train,
  est_data=est_data,
  est_treatment=est_data$W_train,
  minsize=1,                  # Min. number of treatment and control cases in each leaf
  HonestSampleSize=nrow(est_data), #  Num obs used in estimation after splitting
  # We recommend not changing the parameters below
  split.Rule="CT",            # Define the splitting option
  cv.option="TOT",            # Cross validation options
  cp=0,                       # Complexity parameter
  split.Honest=TRUE,          # Use honesty when splitting
  cv.Honest=TRUE              # Use honesty when performing cross-validation
)
```

```
## [1] 2
## [1] "CT"
```

```
unpruned = ct.unpruned

ct.cptable <- as.data.frame(ct.unpruned$cptable)
cp.selected <- which.min(ct.cptable$xerror)
cp.optimal <- ct.cptable[cp.selected, "CP"]

# Prune the tree at optimal complexity parameter.
ct.pruned <- prune(tree=ct.unpruned, cp=cp.optimal)

# Predict point estimates (on estimation sample)
tau.hat.est <- predict(ct.pruned, newdata=test_data)
```

```r
# Create a factor column 'leaf' indicating leaf assignment in the estimation set
num.leaves <- length(unique(tau.hat.est))
leaf <- factor(tau.hat.est, levels=sort(unique(tau.hat.est)), labels = seq(num.leaves))

# Plot unpruned
rpart.plot(
  x=ct.unpruned,      # Pruned tree do ct.pruned, else ct.unpruned
  type=3,             # Draw separate split labels for the left and right directions
  fallen=TRUE,        # Position the leaf nodes at the bottom of the graph
  leaf.round=1,       # Rounding of the corners of the leaf node boxes
  extra=100,          # Display the percentage of observations in the node
  branch=.1,          # Shape of the branch lines
  box.palette="GnPu") # Palette for coloring the node
```




```r
# Plot pruned tree
rpart.plot(
  x=ct.pruned,      # Pruned tree do ct.pruned, else ct.unpruned
  type=3,             # Draw separate split labels for the left and right directions
  fallen=TRUE,        # Position the leaf nodes at the bottom of the graph
  leaf.round=1,       # Rounding of the corners of the leaf node boxes
  extra=100,          # Display the percentage of observations in the node
  branch=.1,          # Shape of the branch lines
  box.palette="GnPu") # Palette for coloring the node
```


0.11
100%