Lab 6 - Machine Learning for Social Science

To be handed in no later than October 15th, 10:00. The submission should include code, relevant output, as well as answers to questions. We recommend the use of RMarkdown to create the report.

Part 1: Meta-learners for job training evaluation

The dataset `"job_training_updated.csv"` contains information about 12,000 individuals who either participated or did not participate in a job training program, including:

- `training`: Binary indicator of whether individual participated in training (treatment)
- `earnings`: Post-training annual earnings in thousands of dollars (outcome)
- `age`: Age of individual
- `education`: Years of education
- `prior_earnings`: Earnings before training program
- `employment_history`: Years of prior employment
- `urban`: Binary indicator of urban residence

```
library(data.table)
library(rpart)
library(randomForest)
library(caret)
library(htetree)
library(rpart.plot)
library(grf)
library(ggplot2)
```

```
# Load CSVs.
setwd('/Users/marar08/Documents/Teaching/MLSS_HT2025/Labs/W6/toupload/')
jt <- fread("/Users/marar08/Documents/Teaching/MLSS_HT2025/Labs/W6/toupload/job_training_updated.csv")
schl <- fread("/Users/marar08/Documents/Teaching/MLSS_HT2025/Labs/W6/toupload/scholarship.csv")
```

1. Fit regular OLS regression using `lm()`, including all non-treatment and non-outcome variables as control variables. Interpret the coefficient for the treatment variable as the average treatment effect. Considering what we talked about in the lecture, what properties of the data would lead you to believe your estimate is biased? Motivate.

```
ols1 <- lm(earnings ~ training + age + education + prior_earnings +
                      employment_history + urban + skill,
           data = jt)
summary(ols1)$coef["training", , drop = FALSE]
```

```
##          Estimate Std. Error  t value      Pr(>|t|)
## training 8.381555  0.3493847 23.98947 2.952623e-124
```

The OLS-estimate suggests that attending the job training program increases ones earnings by approximately 8,400 dollars. However, if the confounders (e.g., age, education) are non-linearly associated with the treatment (attending the job training program), a standard OLS model with those confounders included in the model will not ensure "like-with-like" comparisons, potentially biasing the causal effect.

2. Next, you shall estimate an orthogonal learner, using decision trees as the method for predicting both the treatment and the outcome. Please follow the following steps:

   a. Train a decision tree model using `rpart()` to predict `training` from all confounders using the full dataset. For classification trees, use `method="class"` and for the control parameters use: `cp=0, minbucket=5, maxdepth=30` (i.e., `control=rpart::rpart.control(cp=0,minbucket=5, maxdepth=30)`).
   b. Train a decision tree model using `rpart()` to predict `earnings` from all confounders using the full dataset. For regression trees, use `method="anova"` and the same control parameters: `cp=0, minbucket=5, maxdepth=30` (i.e., `control=rpart::rpart.control(cp=0,minbucket=5, maxdepth=30)`).
   c. Make predictions of treatment (using model from a, with `type="prob"`) and outcome (using model from b) for all observations.
   d. Calculate residuals for all observations: `X_tilde = X - X_hat`, `Y_tilde = Y - Y_hat`.
   e. Estimate the ATE by regressing `Y_tilde` on `X_tilde` using `lm()`.
   f. Report the ATE. How does it compare to your OLS estimate in #1?
   g. Which of the two methods do you trust more? Can you think of any aspect of the implementation of the orthogonal learner which could bias its estimate?

```r
# Train and predict treatment using full sample and classification tree
t_x <- rpart(training ~ age + education + prior_earnings +
                        employment_history + urban + skill,
             data = jt,
             method = "class",
             control = rpart.control(cp = 0,
                                     minbucket = 5,
                                     maxdepth = 30))
mhat <- predict(t_x, type = "prob")[,"1"]

# Train and predict outcome using full sample and regression tree
t_y <- rpart(earnings ~ age + education + prior_earnings +
                        employment_history + urban + skill,
             data = jt,
             method = "anova",
             control = rpart.control(cp = 0,
                                     minbucket = 5,
                                     maxdepth = 30))
ghat <- predict(t_y)

# Calculate residuals
jt[, `:=`(X_tilde = training - mhat,
          Y_tilde = earnings - ghat)]

# Regress outcome residuals on treatment residuals
orth_tree <- lm(Y_tilde ~ X_tilde, data = jt)
summary(orth_tree)$coef["X_tilde", , drop = FALSE]
```

```
##         Estimate Std. Error  t value     Pr(>|t|)
## X_tilde 3.542163   0.212509 16.66829 1.098804e-61
```

Using this approach, our estimate effect of the job training program is substantially lower: 3,500 dollars compared to 8,400 dollars. Which do we trust more? This is a non-trivial question, and depends on (a) how non-linear we think the counfounding relations are, and (b) how overfitted we suspect our models in the orthogonal learner could be: if they are overfitted, and we estimate the treatment effect on the same data as we trained the prediction models on, this can cause a leakage whereby residuals become spuriously small, biasing the treatment effect – generally downwards. In our case, we are not pruning our trees (cp=0), and allow them to grow large (minbucket=5, maxdepth=30), suggesting we are likely to be overfitting them, and may therefore bias our estimated treatment effect.

3. Given your conclusions in #2, do you think either of the following two changes to the setup of the orthogonal learner could improve the ATE estimate? (i) switching from a decision tree to a random forest, (ii) add cross-fitting. Motivate.

(i) Random forests, as we have learnt in the course, reduce variance and capture nonlinearity more stably than a single tree: i.e., we should be less likely to be overfit our prediction models. (ii) Cross-fitting breaks the within-sample dependence between prediction model fits and residualization – ensuring that problems of overfitting does not spill over to the estimated treatment effects. In other words, we should expect our prediction models to be less likely to overfit, and for overfitting that occurs, its negative downstream effects on estimated treatment effect is limited.

4. Now you shall implement the two updates discussed in #3. Please do the following:

   a. Divide your data into 5 folds (hint: you can use `createFolds()` from the `caret` package).
   b. Create a for-loop which in each iteration i does the following:
      i. Train a random forest model using `randomForest()` (with `ntree=200` and `mtry=2`) predicting `training` from confounders on data in folds $\neq i$.
      ii. Train a random forest model using `randomForest()` (with `ntree=200` and `mtry=2`) predicting `earnings` from confounders on data in folds $\neq i$.
      iii. Use models from (i) and (ii) to predict treatment (with `type="prob"`) and outcome for observations in fold i.
      iv. Calculate residuals `X_tilde` and `Y_tilde` for observations in fold i.
      v. Store residuals from fold i.
   c. Combine dataset of residuals and regress `Y_tilde` on `X_tilde` using `lm()`.
   d. Report the estimated ATE. Do you trust this estimate more than those in #2, and if so why (or why not)?

```r
# To keep code clean, vector of input variables
covars <- c("age","education","prior_earnings",
            "employment_history","urban","skill")
# Divide into folds
set.seed(1)
folds <- caret::createFolds(jt$training, k = 5, returnTrain = FALSE)
# Vectors to store output
xhat_oof <- yhat_oof <- rep(NA_real_, nrow(jt))
# Cross-fitting
for (i in seq_along(folds)) {
  te <- folds[[i]];
  tr <- setdiff(seq_len(nrow(jt)), te)
```

```r
  # Estimate models on all folds except i'th
  rf_x <- randomForest(x = as.data.frame(jt[tr, ..covars]),
                       y = as.factor(jt$training[tr]),
                       ntree=200,
                       mtry=2)
  rf_y <- randomForest(x = as.data.frame(jt[tr, ..covars]),
                       y = jt$earnings[tr],
                       ntree=200,
                       mtry=2)
  # Predict on i'th fold
  xhat_oof[te] <- predict(object = rf_x,
                          newdata = as.data.frame(jt[te, ..covars]),
                          type="prob")[,"1"]
  yhat_oof[te] <- predict(object = rf_y,
                          newdata = as.data.frame(jt[te, ..covars]))
}
# Calculate residuals
jt[, `:=`(X_tilde = training - xhat_oof,
          Y_tilde = earnings - yhat_oof)]
# Regress outcome residual on the treatment residual
orth_forest <- lm(Y_tilde ~ X_tilde, data = jt)
summary(orth_forest)$coef["X_tilde", , drop = FALSE]
```

```
##         Estimate Std. Error t value      Pr(>|t|)
## X_tilde 5.679244  0.2568713 22.1093 3.344114e-106
```

Combining random forests and cross-fitting, our estimate of the effect of the job training program falls inbetween the previous two estimates: 5,700 dollars. This approach addresses both of the concerns outlined in #3: (i) it captures non-linerity and is therefore superior to the OLS-approach in case the confounding relation is non-linear, and (ii) it reduces the risk of overfitting due to the use of random forest instead of decision trees, while also limiting the impact of any overfitting on the final treatment effect estimates by using cross-fitting.

5. Suppose we learn that the true average treatment effect is 5.5 thousand dollars. Report which method came closest, and discuss what this says about the properties of the data—in particular the relation between the confounders and the treatment and outcome.

```r
res_tbl <- data.table(
  method = c("OLS", "Orthogonal learner (trees, in-sample)", "Orthogonal learner (RF, 5-fold CF)"),
  ATE = c(coef(ols1)["training"], coef(orth_tree)["X_tilde"], coef(orth_forest)["X_tilde"]),
  true_ATE = 5.5
)
res_tbl
```

```
##                                   method      ATE true_ATE
##                                   <char>    <num>    <num>
## 1:                                   OLS 8.381555      5.5
## 2: Orthogonal learner (trees, in-sample) 3.542163      5.5
## 3:    Orthogonal learner (RF, 5-fold CF) 5.679244      5.5
```

The approach that comes closest to the ground truth is the orthogonal learner with cross-fitting and using random forest. The fact that we se an improvement compared to the standard OLS approach suggests that there is indeed a non-linear confounding relationship. The improvement over the orthogonal learner without cross-fitting and using decision trees in turn suggests that the latter likely was overfitted.

Part 2: Heterogeneity I

The dataset `"scholarship.csv"` contains information about 15,000 students who either received or did not receive a college scholarship, including:

- `scholarship`: Binary indicator of scholarship receipt (treatment)
- `completed`: Binary indicator of degree completion within 6 years (outcome)
- `gpa`: High school GPA (scale 0-4)
- `parental_income`: Parental income in thousands of dollars
- `first_generation`: Binary indicator of first-generation college student status
- `sat_score`: SAT score (scale 400-1600)
- `distance_to_college`: Distance from home to college in miles
- `financial_need`: Measure of financial need (scale 0-100)

1. Suppose your co-author, who has done a careful literature review, has found support for two of the variables, `first_generation` and `financial_need`, having a moderating effect. What you shall do first is examine whether you find evidence of this in your data. Implement a standard linear regression using `lm()` (or `glm()` if you prefer logistic regression) with the treatment variable as well as all other input variables (presumed confounders) included, with `first_generation` and `financial_need` interacted with the treatment variable. Report your findings: do you find evidence supporting your colleague's conclusion from the literature?

```r
logit1 <- glm(completed ~ scholarship * first_generation +
                          scholarship * financial_need +
                          gpa + parental_income + sat_score + distance_to_college,
              data = schl, family = binomial())
summary(logit1)$coef
```

```
##                                 Estimate   Std. Error    z value      Pr(>|z|)
## (Intercept)                 -4.687901054 0.2476398770 -18.930316  6.417661e-80
## scholarship                 -0.256384023 0.1304941737  -1.964716  4.944709e-02
## first_generation            -0.095967946 0.0787965106  -1.217921  2.232539e-01
## financial_need              -0.002758877 0.0022169396  -1.244453  2.133330e-01
## gpa                          1.279380695 0.0453439382  28.215033 3.824623e-175
## parental_income              0.003667349 0.0005017991   7.308400  2.703420e-13
## sat_score                    0.001567777 0.0001395628  11.233487  2.792350e-29
## distance_to_college         -0.005849744 0.0010942132  -5.346073  8.988290e-08
## scholarship:first_generation 1.580471099 0.1077557151  14.667167  1.046225e-48
## scholarship:financial_need   0.010570230 0.0026615900   3.971397  7.145248e-05
```

In support of the idea that there is a moderating effect by first_generation and financial_need, we find that the interaction effects are statistically significant for both.

2. Considering what we discussed in the lecture, what is one limitation of this standard approach to effect heterogeneity? What are properties of the data (or state of the field) that could make this limitation more or less problematic?

A key limitation of the standard approach to effect heterogeneity is that it presumes we know beforehand which (combinations of) variables are moderating the treatment effect. Thus, if we have a lot of potential variables that could moderate the effect, and the field remains unsettled which are the important ones, the standard approach is not ideal.

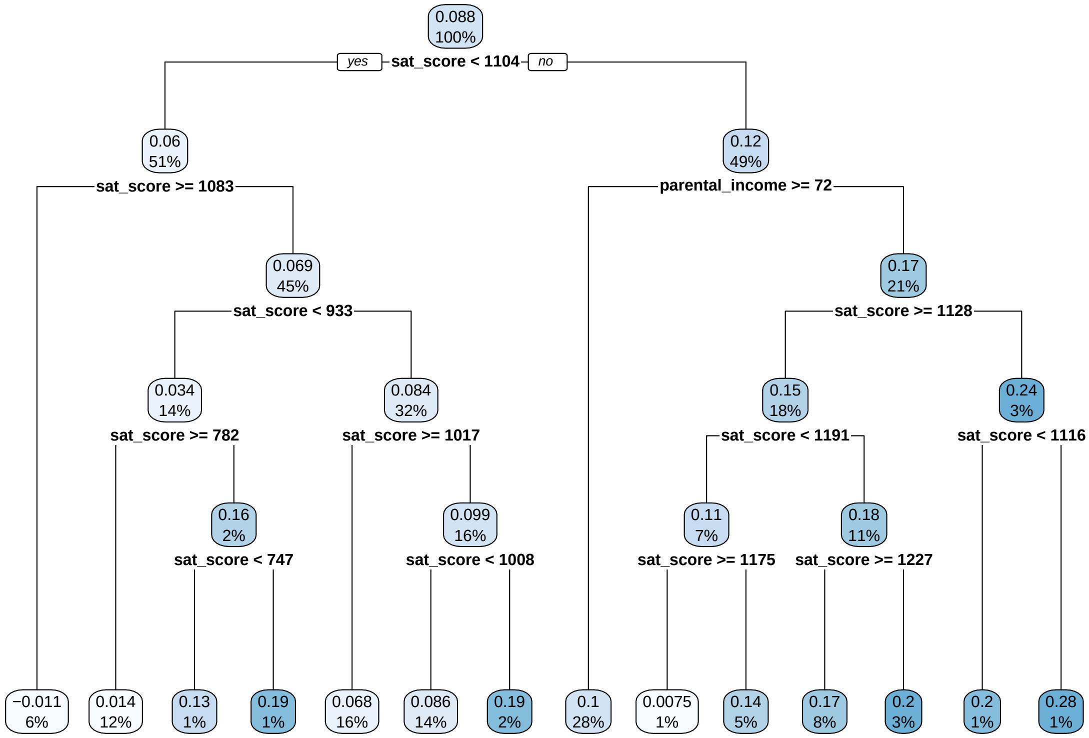
3. Next, you shall consider an alternative approach to effect heterogeneity, using causal trees. At a high level, describe what is the key difference in assumption we make when using causal trees compared to the traditional approach?

In both causal trees and in the standard approach, we make the assumption that we don't have any unobserved confounders. A key difference between the two is that while the standard approach requires the analyst to specify (assume beforehand) which effects are moderating the treatment effect, the causal tree approach learns this from the data. The causal tree seeks splits which maximize heterogeneity in treatment effects.

4. Perform a causal tree analysis by doing the following:

   a. Estimate the causal tree using the function `causalTree()` from the `htetree` package, specifying the formula as in #1 (except drop the interactions and leave out the treatment variable; the latter is specified separately). Use the following parameters: `split.Rule="CT"`, `cv.option="CT"`, `split.Honest=TRUE`, `split.Bucket=TRUE`, `minsize=60`, `cp=0`, `bucketNum=40`.
   b. Visualize the tree using `rpart.plot()` and describe the combination of splits which identify the population with (a) the largest treatment effect and (b) the smallest.
   c. Suppose our dataset is a standard observational dataset common to the social sciences, e.g., a survey dataset of a random sample of the population. Given this information, what could be a potential threat to the validity of our causal tree results?

```r
# a) Fit the causal tree
set.seed(42)
ct <- causalTree(
  completed ~ gpa + parental_income + first_generation +
              sat_score + distance_to_college + financial_need,
  data = as.data.frame(schl),
  treatment = schl$scholarship,
  split.Rule   = "CT",
  split.Honest = TRUE,
  cv.option    = "CT",
  cv.Honest    = TRUE,
  split.Bucket = TRUE,
  bucketNum    = 40,
  minsize      = 60,
  cp           = 0
)
# b) Visualize the tree
rpart.plot::rpart.plot(x = ct)
```



The strongest positive effect of the treatment is found for the subgroup of students with an SAT score between 1116 and 1128 whose parents income are below 72 thousand dollars (+.28). The smallest (even negative) treatment effect of this scholarship is found for students with an SAT score between 1083 and 1104 (-.01).

In the causal tree we just estimated, we did not account for any confounding selection effects. It may be that certain subsets of individuals just happen to have lower / higher completion rates.

5. Given potential concerns of selection bias, you shall next examine the potential imbalance of treated and untreated observations inside different leaves. To do so, please follow these steps (Hint: various code-chunks are provided that may be helpful):

   a. Estimate a propensity score model—a standard logistic regression model using `glm()` with `family=binomial()`—predicting the treatment variable based on the confounders. Specify `type="response"` in the `predict()` function.

```
# Fit propensity score model
ps_model <- glm(scholarship ~ gpa + parental_income + first_generation + sat_score +
                  distance_to_college + financial_need,
                data = schl, family = binomial())
# Predict propensity scores
schl[, ps_hat := as.numeric(predict(ps_model, type="response"))]
```

b. Calculate the mean and standard deviation of the propensity scores within each leaf and treatment group combination. (Hint: use `$where` to extract leaf assignments)

```r
# Extract leaf ids and add to data.table
leaves <- factor(ct$where)
schl[, leaf := leaves]
```

```r
# Calculate means, sds
leaf_group_long <- schl[, .(
  n       = .N,
  mean_ps = mean(ps_hat),
  sd_ps   = sd(ps_hat)
), by = .(leaf, scholarship)]
leaf_group_stats <- data.table::dcast(
  leaf_group_long,
  leaf ~ scholarship,
  value.var = c("n", "mean_ps", "sd_ps"),
  fill = NA_real_
)
```

c. Based on the mean and standard deviation, calculate the standardized difference in means measure within each leaf. What do these indicate about your results in #4?

```r
# Calculate SMD
leaf_balance <- copy(leaf_group_stats)[
  , `:=`(
      SMD_ps = {
        denom <- sqrt((sd_ps_1^2 + sd_ps_0^2)/2)
        ifelse(is.finite(denom) & denom > 0,
               abs(mean_ps_1 - mean_ps_0) / denom, NA_real_)
      }
    )
][order(-SMD_ps)]
print(leaf_balance)
```

```
##       leaf   n_0   n_1 mean_ps_0 mean_ps_1   sd_ps_0   sd_ps_1    SMD_ps
##     <fctr> <int> <int>     <num>     <num>     <num>     <num>     <num>
##  1:      3   404   493 0.4706886 0.6578065 0.1914150 0.1697804 1.0342493
##  2:      9    59    72 0.4583521 0.6614164 0.2187141 0.1814856 1.0104523
##  3:     24   128   320 0.6060854 0.7408130 0.1415111 0.1325363 0.9827162
##  4:      8    64    69 0.4552790 0.6412623 0.1912523 0.2045044 0.9393606
##  5:     11   991  1391 0.4824593 0.6530247 0.1922244 0.1714161 0.9365672
##  6:      6   746  1023 0.4689748 0.6397324 0.1952731 0.1805348 0.9080514
##  7:     13   892  1208 0.4780980 0.6424285 0.1983657 0.1760932 0.8761477
##  8:     26    63   136 0.5934024 0.7229937 0.1625895 0.1372916 0.8612253
##  9:     16  2040  2140 0.4312543 0.5915192 0.1941737 0.1847191 0.8457009
## 10:     21   240   529 0.6064419 0.7208652 0.1434740 0.1437543 0.7967407
## 11:     14   133   161 0.5078858 0.6590003 0.2034602 0.1760142 0.7943662
## 12:     23   396   876 0.6139195 0.7170038 0.1467418 0.1372404 0.7255853
## 13:     20    60   161 0.6158418 0.7105146 0.1558878 0.1339385 0.6514415
## 14:     27    68   137 0.6354051 0.7149636 0.1551393 0.1396181 0.5390773
```

The standardized mean differences (SMD) for all leaves are substantially greater than conventional thresholds for decent balance (i.e., 0.10, 0.25), suggesting that there seems to be meaningful (unaccounted for) selection into different leaves; and treated and untreated in the leaves are not comparable. In other words, the estimated treatment effects in the different leaves may thus just reflect baseline differences between treated and untreated — without having anything to do with the effect of the treatment itself.

6. Given your findings in the previous task, you shall next do a causal tree analysis wherein you incorporate inverse probability weighting. To do so, please do the following:

    a. Refit the causal tree using `causalTree()` with same specifications as in #4, and add the `weights` argument set to `1/p` for treated units and `1/(1-p)` for control units, where `p` is the predicted propensity score (see code chunk below for how you could do this). This incorporates IPW into the tree.

```r
# Crate IPW
schl$w_ipw <- ifelse(test = schl$scholarship==1,
                     yes = 1/pmax(schl$ps_hat, 0.02),
                     no = 1/pmax(1-schl$ps_hat, 0.02))
```

```r
# Fit causal tree
set.seed(43)
ct3 <- causalTree(
    completed ~ gpa + parental_income + first_generation +
                sat_score + distance_to_college + financial_need,
    data = as.data.frame(schl),
    treatment = schl$scholarship,
    weights = schl$w_ipw,
    split.Rule   = "CT",
    split.Honest = TRUE,
    cv.option    = "CT",
    cv.Honest    = TRUE,
    split.Bucket = TRUE,
    bucketNum    = 40,
    minsize      = 60,
    cp           = 0)
```

b. Assess the balance for this tree in the same way you did in #5 (but you can skip the first step which estimates the propensity score model). Did the balance improve in comparison to #4?

```r
# Assess balance
leaves3 <- factor(ct3$where)
schl[, leaf_ct3 := leaves3]

leaf_group_long3 <- schl[, .(
  n       = .N,
  mean_ps = weighted.mean(ps_hat,w = w_ipw), # OBS!
  sd_ps   = sd(ps_hat)
), by = .(leaf_ct3, scholarship)]

leaf_group_stats3 <- data.table::dcast(
  leaf_group_long3,
  leaf_ct3 ~ scholarship,
  value.var = c("n", "mean_ps", "sd_ps"),
  fill = NA_real_
)
```

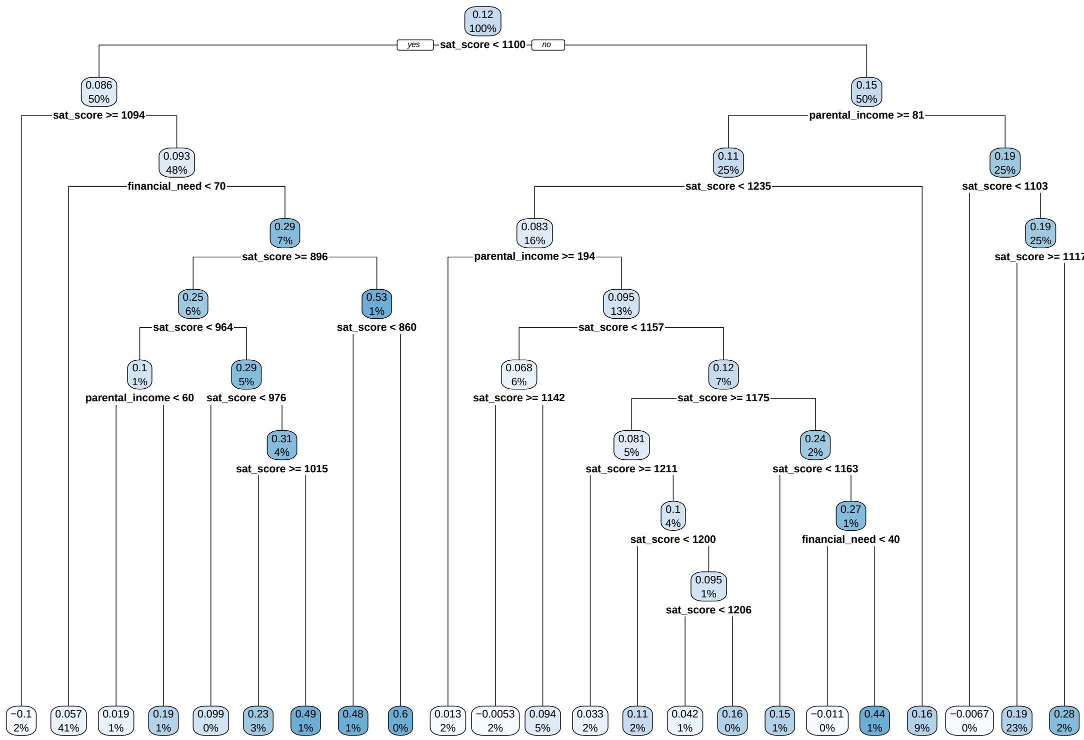
```r
leaf_balance3 <- copy(leaf_group_stats3)[
  , `:=`(
      SMD_ps = {
        denom <- sqrt((sd_ps_1^2 + sd_ps_0^2)/2)
        ifelse(is.finite(denom) & denom > 0,
               abs(mean_ps_1 - mean_ps_0) / denom, NA_real_)
      }
    )
][order(-SMD_ps)]
print(leaf_balance3)
```

```
##     leaf_ct3   n_0   n_1 mean_ps_0 mean_ps_1    sd_ps_0    sd_ps_1      SMD_ps
##       <fctr> <int> <int>     <num>     <num>      <num>      <num>       <num>
##  1:        9    15   113 0.7995430 0.8419125 0.06713080 0.05834280 0.673702622
##  2:       36    39    47 0.4569035 0.5597650 0.15772965 0.15862835 0.650282648
##  3:       17    16   110 0.8081855 0.8326292 0.05376492 0.06986641 0.392115833
##  4:        3   131   139 0.5494399 0.6154712 0.18904883 0.16089632 0.376165619
##  5:       39    43    65 0.6700520 0.6183779 0.13311139 0.14364469 0.373156718
##  6:       33    41    52 0.5588033 0.4958420 0.17627934 0.20022674 0.333775838
##  7:       18    11    63 0.8507098 0.8337294 0.08161838 0.05702280 0.241188396
##  8:       34    28    49 0.4939536 0.5266024 0.15663303 0.18555560 0.190145172
##  9:       14    64   373 0.8507940 0.8408074 0.06572954 0.06344402 0.154598047
## 10:       26   308   373 0.5478089 0.5279854 0.17423639 0.17141313 0.114698825
## 11:       29   128   129 0.5356555 0.5162736 0.17435209 0.17356795 0.111415511
## 12:       22   256    90 0.2531814 0.2363504 0.13406202 0.17106770 0.109517782
## 13:       15    30   158 0.8464513 0.8388764 0.08666559 0.06810320 0.097190514
## 14:       31   170   198 0.5558429 0.5398952 0.17622265 0.17457829 0.090920779
## 15:       38    48    25 0.3642354 0.3737500 0.11989502 0.09666210 0.087369543
## 16:       10    16    74 0.8345108 0.8312460 0.03612886 0.05293073 0.072046807
## 17:       12    12    45 0.8491378 0.8463772 0.05123759 0.05301073 0.052954353
## 18:       45    97   186 0.6921625 0.6862635 0.16194821 0.14041631 0.038920309
## 19:       40   706   675 0.5058907 0.4996980 0.19052903 0.18299866 0.033151032
## 20:       25   107   138 0.5438780 0.5484892 0.16284102 0.17292125 0.027455022
## 21:        5  2921  3242 0.5295975 0.5275020 0.18319265 0.15624822 0.012308099
## 22:       42    20    44 0.6569944 0.6582726 0.14966116 0.14581637 0.008651011
## 23:       44  1077  2328 0.6799972 0.6805879 0.14688506 0.14117310 0.004100811
##     leaf_ct3   n_0   n_1 mean_ps_0 mean_ps_1    sd_ps_0    sd_ps_1      SMD_ps
```

Here I realize I left out a crucial piece of information. When calculating the balance, the means that we calculate should be weighted based on the IPW weights. Otherwise we don't take into account the fact that we have estimated a weigthed causal tree: observations contribute to the estimates in proportion to their IPW weights. But ok, so what do we find with appropriate weighthing? The balance has substantially improved. Now a majority of the leaves have balance below the conventional thresholds of 0.25 and 0.10. Some leaves are still imbalanced, so we should be a bit extra cautious when interpretting those.

c. Visualize the tree and provide an interpretation of its structure, highlighting what you think is interesting in it. Are the conclusions you draw from this tree different from those in #4? What does this suggest about the findings in #4?

```r
# Visualize tree
rpart.plot::rpart.plot(ct3)
```



One similarity is that SAT score appear frequently in the tree here as well. One notable difference is that here — compared to the previous causal tree — variables like ‘financial_need‘ and ‘parental_income‘ are featuring relatively promenantly in the tree. For example, the subgroup with the largest treatment effect here is: $860 \leq SAT \leq 895$ and $financial\_aid < 70$.

d. To get a sense of the subgroups contained within each leaf (of interest), please describe its average properties in terms of all the input variables (except treatment).

```
# Extract estimates from frame
fr        <- ct3$frame
leaf_nums <- as.integer(rownames(fr))
leaf_est   <- data.table(leaf = leaf_nums[fr$var=="<leaf>"],
                         leaf_estimate = fr$yval[fr$var=="<leaf>"])
# Add to data.table
schl[, frame_row := ct3$where]
schl[, leaf      := leaf_nums[frame_row]]
# Per-leaf means of covariates
covars <- c('gpa','parental_income','first_generation','sat_score',
             'distance_to_college','financial_need')
leaf_means <- schl[, lapply(.SD, mean, na.rm = TRUE),
                   .SDcols = covars,
                   by = leaf]
# Combine with estimate
leaf_means_est <- merge(x=leaf_means,
                        y=leaf_est,
                        by='leaf')
```

```r
# Print
print(leaf_means_est[order(leaf_estimate,decreasing = T)])
```

```
##      leaf      gpa parental_income first_generation sat_score
##     <int>    <num>           <num>            <num>     <num>
##  1:    47 2.993615        64.07703        0.8243243  878.1892
##  2:   183 2.984262        57.95000        0.7978723  997.0638
##  3:    46 2.959384        59.76349        0.7936508  789.5238
##  4:   415 2.945474       116.61019        0.5185185 1168.5926
##  5:    31 2.967937        52.53958        0.3639576 1108.7279
##  6:   182 2.955822        58.80549        0.7757437 1055.6842
##  7:    30 3.002069        52.43924        0.3389134 1230.4552
##  8:    89 2.984705        88.43333        0.9333333  929.5667
##  9:   823 3.025579       116.68052        0.4285714 1208.0130
## 10:    13 2.970810       140.44873        0.3461260 1315.9819
## 11:   206 2.976392       122.20116        0.3372093 1159.9767
## 12:   410 2.984648       120.80788        0.3885870 1187.3777
## 13:    90 3.085043        58.01754        0.9122807  969.3684
## 14:   101 2.982261       119.07606        0.3700441 1120.1542
## 15:    10 2.997334       103.21426        0.2678890  975.2521
## 16:   822 2.953529       117.35376        0.3333333 1202.3226
## 17:   204 3.004071       118.21401        0.3112840 1222.8833
## 18:    88 2.990017        40.60469        0.7031250  931.6562
## 19:    24 2.973772       265.16040        0.3265896 1161.5578
## 20:   100 2.992646       117.95102        0.3714286 1148.7388
## 21:    14 2.958443        54.33594        0.3593750 1101.0000
## 22:   414 3.047101       131.43151        0.1369863 1168.8219
## 23:     4 3.035271       102.57222        0.4222222 1096.5074
##      leaf      gpa parental_income first_generation sat_score
##     distance_to_college financial_need leaf_estimate
##                   <num>          <num>         <num>
##  1:            33.69324       78.02703   0.600221374
##  2:            32.11170       78.94681   0.492665265
##  3:            27.80238       78.03968   0.477276578
##  4:            30.48426       54.37963   0.437167929
##  5:            30.12120       59.25088   0.281692619
##  6:            30.77529       78.78719   0.227584175
##  7:            30.15289       59.61527   0.186101176
##  8:            28.45000       76.82222   0.185504745
##  9:            31.04545       45.98701   0.157430998
## 10:            30.20442       40.94786   0.155367445
## 11:            29.05814       42.91860   0.150249029
## 12:            30.43288       46.25543   0.110213481
## 13:            29.52105       79.45614   0.099127675
## 14:            29.16432       45.55947   0.093721274
## 15:            30.58671       45.10433   0.057357428
## 16:            27.94624       45.87097   0.042083634
## 17:            30.91946       44.16732   0.033149828
## 18:            30.16172       78.57031   0.019075770
## 19:            31.87283       17.54335   0.012854483
## 20:            29.80898       47.65714  -0.005275149
## 21:            31.05937       57.09375  -0.006742662
## 22:            33.04795       29.38356  -0.010616206
```

```
## 23:            30.77037       49.55185  -0.103075194
##     distance_to_college financial_need leaf_estimate
```

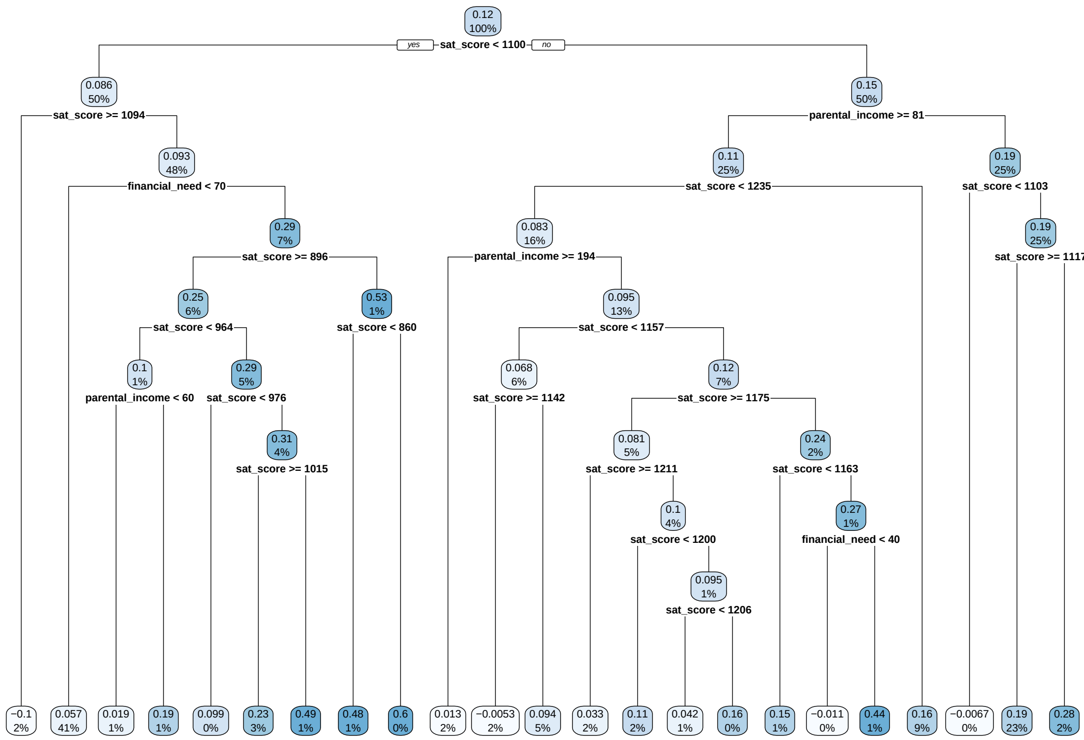
Let's consider the leaves with the largest and the weakest treatment effect respectively. The leaf capturing the largest treatment effect (+.6) contains individuals with an average greater financial need (average: 78), distance to college (average: 34) and more first generation (82%) than many of the other leafs/subsets. The subset which experience the smallest effect (-.1), by contrast, are characterized by lower financial need and higher gpa — again, compared to many of the other leafs/subsets.

e. How do these results map onto your findings in the first analysis in #1? Do you find that the variables suggested are most important indeed are so? What would you say to your co-author?

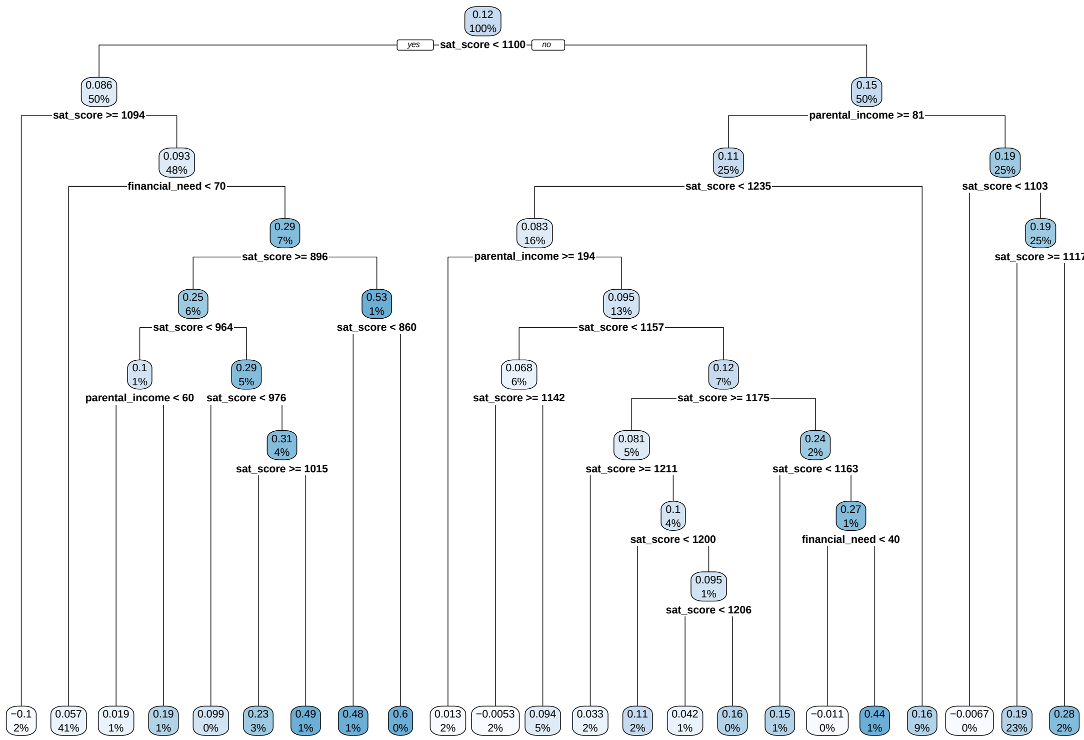
Several of the leafs which display larger treatment effects are characterized by higher financial need; so this aspect I would say is in line with (support of) our co-authors literature review. The first generation indicator is a bit less clear: while the leafs with the largest treatment effects have higher than average shares, there are also other leafs with even higher shares that have weaker estimated effects. Beyond what was a priori expected — importantly — the causal tree seems to put a lot of weight (splits frequently) on the SAT score variable. Indeed, if one extracts variable importance from the causal tree, one finds a 5x larger value for the SAT score compared to the second-most important variable.

7. (BONUS) Lastly, to get a sense of the sensitivity of the tree structure to the particular data we estimate it on, please estimate (and visualize) two different trees with different `seed.numbers` assigned. Do they depart meaningfully from each other? Speculate about why you think this difference (or lack thereof) exists.

```r
# Seed 10^1
set.seed(10^0)
ct4 <- causalTree(
    completed ~ gpa + parental_income + first_generation +
                sat_score + distance_to_college + financial_need,
    data = as.data.frame(schl),
    treatment = schl$scholarship,
    weights = schl$w_ipw,
    split.Rule   = "CT",
    split.Honest = TRUE,
    cv.option    = "CT",
    cv.Honest    = TRUE,
    split.Bucket = TRUE,
    bucketNum    = 40,
    minsize      = 60,
    cp           = 0)
rpart.plot::rpart.plot(ct4)
```




```r
# Seed 10^6
set.seed(10^6)
ct5 <- causalTree(
    completed ~ gpa + parental_income + first_generation +
                sat_score + distance_to_college + financial_need,
    data = as.data.frame(schl),
    treatment = schl$scholarship,
    weights = schl$w_ipw,
    split.Rule    = "CT",
    split.Honest = TRUE,
    cv.option     = "CT",
    cv.Honest     = TRUE,
    split.Bucket = TRUE,
    bucketNum     = 40,
    minsize       = 60,
    cp            = 0)
rpart.plot::rpart.plot(ct5)
```

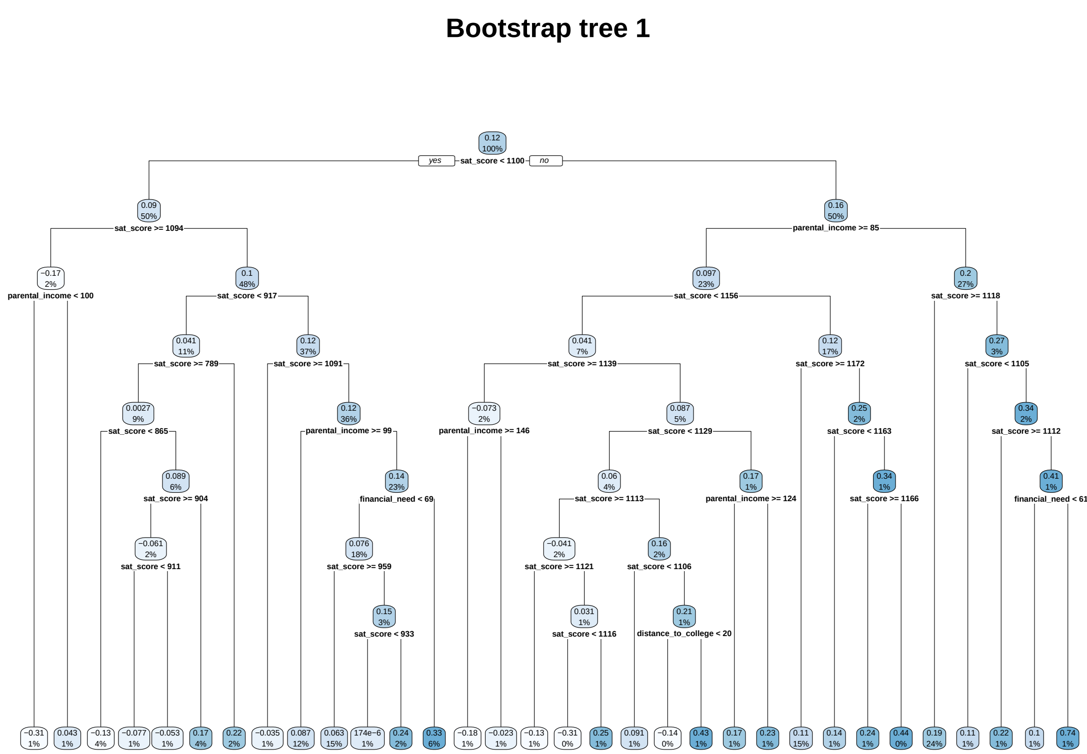


Interestingly, no visible difference between the two? I'm not fully sure exactly why this is. It is true that we are using the same data as input. So, if there is a deterministic way to dividing up the data (which to learn the tree and which to estimate the effect) that is not dependent on an external seed number, then this is the expected result. Seems feasible. Let's therefore try bootstrapping instead.

```
# To reduce copy-pasting, a function for estimating the tree
fit_ct <- function(d) {
  causalTree(
    completed ~ gpa + parental_income + first_generation +
                sat_score + distance_to_college + financial_need,
    data = as.data.frame(d),
    treatment = d$scholarship,
    weights   = d$w_ipw,
    split.Rule   = "CT",
    split.Honest = TRUE,
    cv.option    = "CT",
    cv.Honest    = TRUE,
    split.Bucket = TRUE,
    bucketNum    = 40,
    minsize      = 60,
    cp           = 0
  )
}

# two bootstrap index sets
set.seed(1000)
```

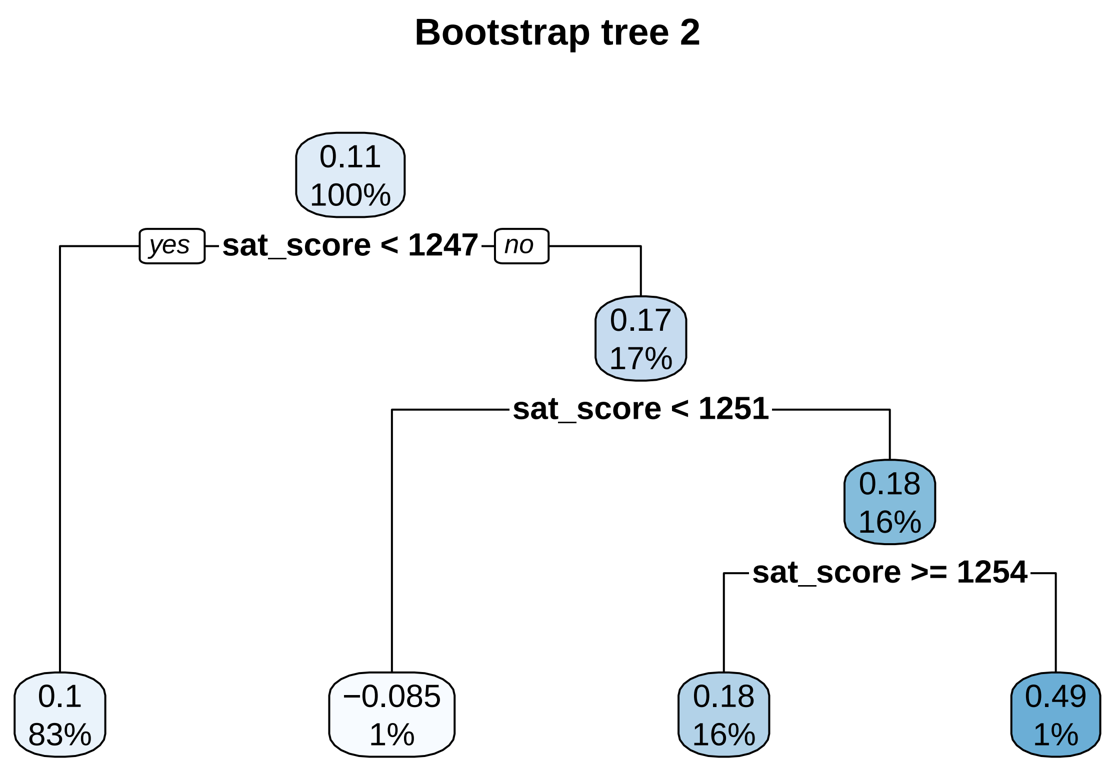
```
n <- nrow(schl)
idx_list <- list(
  sample.int(n, n, replace = TRUE),
  sample.int(n, n, replace = TRUE)
)

# fit causal trees on the two bootstraps
ct_boot <- lapply(idx_list, function(i) fit_ct(schl[i, ]))

# compare plots
rpart.plot::rpart.plot(ct_boot[[1]], main = "Bootstrap tree 1")
```

```
## Warning: Cannot retrieve the data used to build the model (so cannot determine roundint and is.binar
## To silence this warning:
##     Call rpart.plot with roundint=FALSE,
##     or rebuild the rpart model with model=TRUE.
```




```
rpart.plot::rpart.plot(ct_boot[[2]], main = "Bootstrap tree 2")
```

```
## Warning: Cannot retrieve the data used to build the model (so cannot determine roundint and is.binar
## To silence this warning:
##     Call rpart.plot with roundint=FALSE,
##     or rebuild the rpart model with model=TRUE.
```



Just comparing trees estimated on two bootstrap samples, we see a big difference between the resulting trees. This should make us cautious in our interpretations.

Part 3: Heterogeneity II

In this part, you will continue working with the scholarship dataset from Part 2.

1. In this last part, you shall continue with the exploration of heterogeneous treatment effects. But instead of standard OLS and causal trees, you shall use causal forests. Before doing so, please answer the question: why is it not necessary to include inverse probability weighting in causal forest, like we did for trees?

Because causal forests already controls for (non-linear) confounding through the use of orthogonal learning (i.e., predicting the outcome and the treatment based on the confounders, and removing confounding by subtracting those predictions from the original variable).

2. Now, run a causal forest analysis following these steps:

   a. Estimate the causal forest using the function `causal_forest()` from the `grf` package, specifying the input arguments as follows: X (matrix of covariates), Y (outcome), W (treatment), `num.trees=2000`, and `honesty=TRUE`. Set a seed for reproducibility. Report the average treatment effect using `average_treatment_effect()`.

```r
X <- as.matrix(schl[, .(gpa, parental_income, first_generation, sat_score,
                        distance_to_college, financial_need)])
Y <- schl$completed
W <- schl$scholarship
set.seed(50)
cf <- causal_forest(X, Y, W, num.trees = 2000, honesty = TRUE)

# ATE (and SE)
ate <- average_treatment_effect(cf)
```

```
## Warning in average_treatment_effect(cf): Estimated treatment propensities go as
## high as 0.954 which means that treatment effects for some treated units may not
## be well identified. In this case, using ‘target.sample=control‘ may be helpful.
```

```r
print(ate)
```

```
##    estimate     std.err
## 0.115626921 0.008013026
```

b. Examine which variables were most important to account for the heterogeneity in treatment effect by using the function `variable_importance()`. Make a bar chart and interpret. Does this result line up with your findings using causal tree—do the most important variables here overlap with those showing up in the best causal tree?

```r
# Variable importance
vi <- variable_importance(cf)
data.table(var = colnames(X),
           importance = as.numeric(vi))[order(-importance)]
```

```
##                    var importance
##                 <char>      <num>
## 1:    first_generation 0.61358937
## 2:           sat_score 0.10869261
## 3:                 gpa 0.10307270
## 4:      financial_need 0.08212846
## 5: distance_to_college 0.04918010
## 6:     parental_income 0.04333676
```

Here, ‘first_generation‘ is the clear winner; accounting for 6 times as many splits as the second-most important variable (SAT score). So, what is consistent is that the SAT score seems relatively important in both (2nd ranked for the forest; most frequent variable in the tree). However, while ‘first_generation‘ is deemed most important variable for the forest model, it is nowhere to be found in the causal tree. Speculating as to why this might be the case, I would say: (a) we know the causal tree, like a regular tree, have high variance, such that the tree structure that is learnt may importantly depend on the exact data points it is fed; the bootstrap exercise confirmed this for us; while causal forest on the other hand learns many trees and is more stable (b) the two models also apply different approaches to handle confounding; causal forest using state of the art orthogonal learning.

c. For the two variables you identified as most important, please examine how the effects vary along these dimensions. To do so, divide into quintiles of these variables (if continuous) and calculate average treatment effects for each subcategory separately. Plot how the treatment effect varies across quintiles and interpret. Does this result provide additional information to what you could infer from the causal tree?

```r
# Function to identify which observations fall into a give quitile along v
make_rank_quintiles <- function(v) {
  r <- rank(v, ties.method="average", na.last="keep")
  d <- as.integer(ceiling(5 * r / max(r, na.rm=TRUE)))
  d[d < 1] <- 1; d[d > 5] <- 5
  factor(d, levels = 1:5, labels = paste0("Q", 1:5))
  return(d)
}

# Compute CATE by quintiles of SAT
sat_qs <- make_rank_quintiles(schl$sat_score)
bins <- split(seq_len(nrow(schl)), sat_qs)
cate_quint <- rbindlist(lapply(names(bins), function(lbl) {
  idx <- bins[[lbl]]
  est <- average_treatment_effect(cf, subset = idx, target.sample = "all")
  data.table(bin = lbl,
             n = length(idx),
             cate = est[[1]],
             se = est[[2]])}))
```

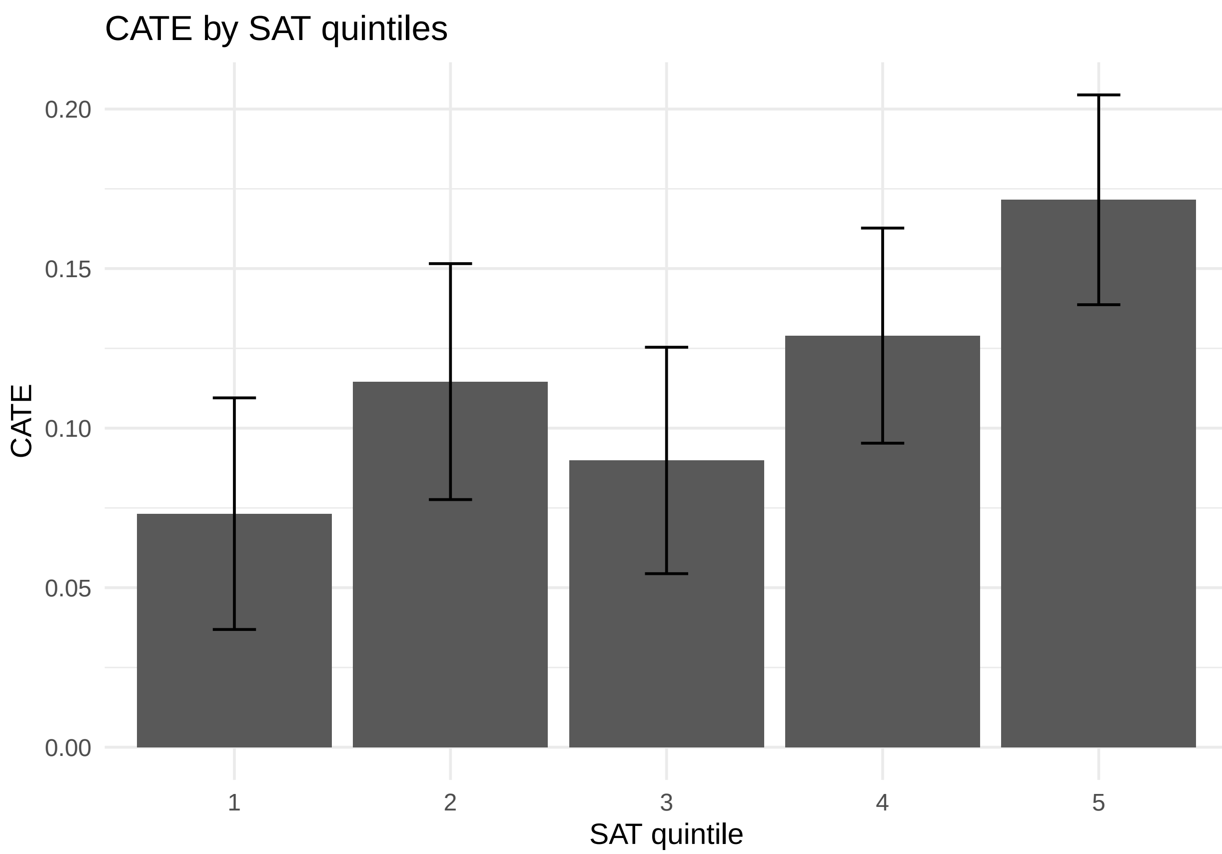
```
## Warning in average_treatment_effect(cf, subset = idx, target.sample = "all"):
## Estimated treatment propensities go as high as 0.954 which means that treatment
## effects for some treated units may not be well identified. In this case, using
## ‘target.sample=control‘ may be helpful.
```

```r
# Compute CATE by first_generation
cate_fg_1 <- average_treatment_effect(forest = cf,
                                      subset = which(schl$first_generation==1),
                                      target.sample = "all")
```

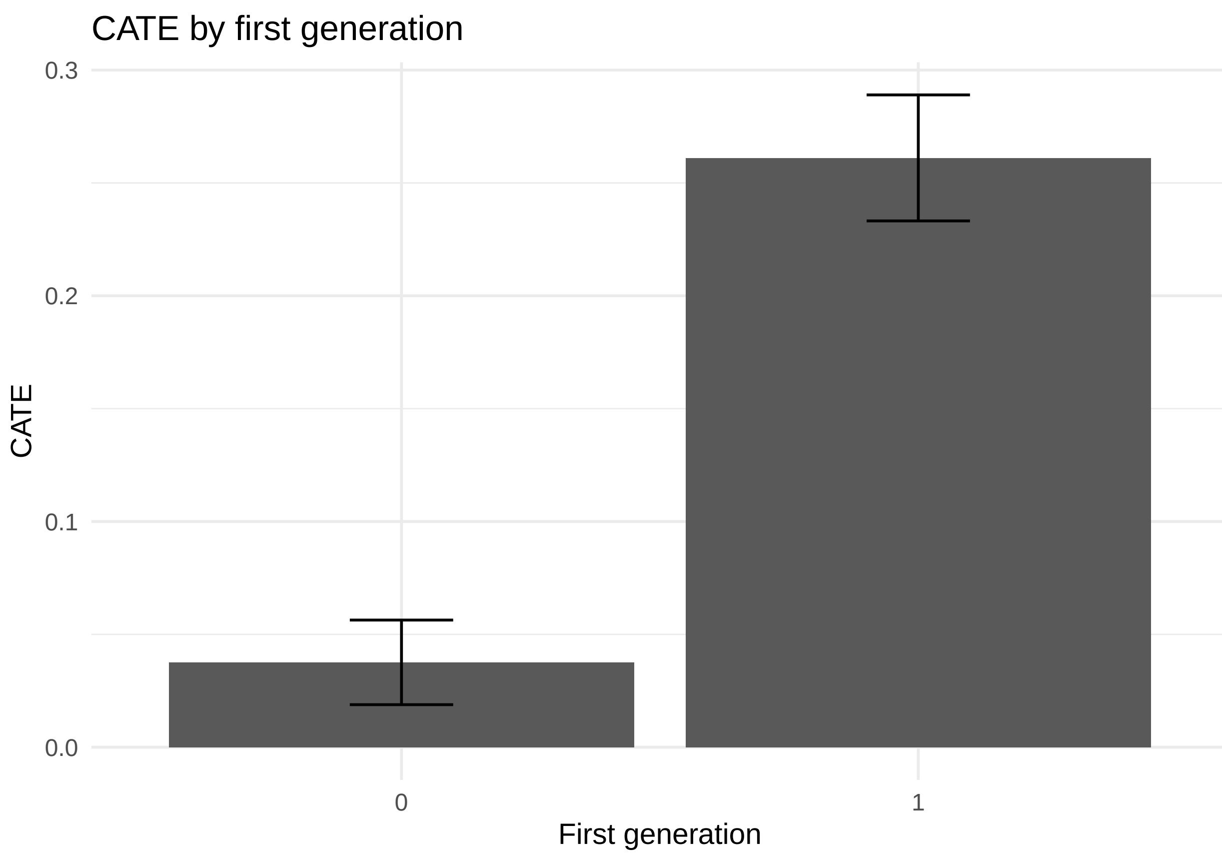
```
## Warning in average_treatment_effect(forest = cf, subset =
## which(schl$first_generation == : Estimated treatment propensities go as high as
## 0.954 which means that treatment effects for some treated units may not be well
## identified. In this case, using ‘target.sample=control‘ may be helpful.
```

```r
cate_fg_0 <- average_treatment_effect(forest = cf,
                                      subset = which(schl$first_generation==0),
                                      target.sample = "all")
cate_fg <- as.data.table(do.call("rbind", list(cate_fg_1,cate_fg_0)))
cate_fg$first_generation <- c(1,0)
```

```r
# Plot CATE by SAT
ggplot(cate_quint, aes(x = bin, y = cate)) +
  geom_col() +
  geom_errorbar(aes(ymin = cate - 1.96*se, ymax = cate + 1.96*se), width = 0.2) +
  labs(x = "SAT quintile", y = "CATE", title = "CATE by SAT quintiles") +
  theme_minimal()
```




```
# Plot CATE by First generation
ggplot(cate_fg, aes(x = factor(first_generation), y = estimate)) +
  geom_col() +
  geom_errorbar(aes(ymin = estimate - 1.96*std.err,
                    ymax = estimate + 1.96*std.err),
                width = 0.2) +
  labs(x = "First generation", y = "CATE", title = "CATE by first generation") +
  theme_minimal()
```


CATE by first generation
0.3
0.2
0.1
0.0
CATE
0
1
First generation