Lab 6

Thomas Haase

October 15, 2025

Table of contents

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(easystats)
library(data.table)
library(kableExtra)

library(rpart)
library(rpart.plot)


library(randomForest)

library(caret)

library(htetree)
library(grf)
```

```r
set.seed(5)

setwd("~/Github/ML-Labs/6")
```

Task 1

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.

```r
d <- data_read("job_training_updated.csv")

m1 <- lm(earnings ~ training + age + education +
         prior_earnings + employment_history + urban, d)

m1 |> report_table() |> summary() |> print_md()
```

Parameter	Coefficient	95% CI	t(11993)	p	Std. Coef.	Fit
(Intercept)	2.35	(-1.81, 6.51)	1.11	0.269	−1.49e-16	
training	24.86	(23.83, 25.89)	47.24	< .001	0.40	
age	−0.19	(-0.34, −0.03)	−2.34	0.019	−0.05	
education	5.86	(5.63, 6.09)	49.37	< .001	0.41	
prior earnings	−0.14	(-0.15, −0.13)	−28.37	< .001	−0.25	
employment history	0.79	(0.64, 0.94)	10.34	< .001	0.24	
urban	2.57	(1.68, 3.47)	5.63	< .001	0.04	

AICc	1.11e+05
R2	0.39
R2 (adj.)	0.39
Sigma	24.40

The ATE of the treatment variable `training` on earnings is very large. Despite the statistical significance the model could be improved by causal modeling. The standard paradigm is problematic because it lacks assumes linearity. In case a confounder is non-linear the estimate will be biased.

Task 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?

```
# a
m_X <- rpart(training ~ age + education + prior_earnings +
               employment_history + urban, d, method = "class",
             control = rpart.control(cp = 0, minbucket = 5,
                                     maxdepth = 30))
# b
m_Y <- rpart(earnings ~ age + education + prior_earnings +
               employment_history + urban, d, method = "anova",
             control = rpart.control(cp = 0, minbucket = 5,
                                     maxdepth = 30))
# c
X_hat <- predict(m_X, newdata = d[3:7], type="prob")[,"1"]
Y_hat <- predict(m_Y, newdata = d[3:7])

# d
residuals <- data.frame(X = d$training - X_hat,
                        Y = d$earnings - Y_hat)

# e
m_ate <- lm(Y ~ X, residuals)

# f
m_ate |> parameters() |> print_md()
```

Parameter	Coefficient	SE	95% CI	t(11998)	p
(Intercept)	−2.54e-15	0.10	(-0.20, 0.20)	−2.46e-14	> .999
X	9.72	0.31	(9.12, 10.33)	31.40	< .001

```
m_ate |> report_parameters(include_intercept = F)
```

```
  - The effect of X is statistically significant and positive (beta = 9.72, 95% CI [9.12,
10.33], t(11998) = 31.40, p < .001; Std. beta = 0.28, 95% CI [0.26, 0.29])
```

Task 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.

Cross-fitting means to predict out-of-fold to block contamination whereby the orthogonal learner does not bias the residuals in hold-out towards 0. To adress the high variance of the trees by prohibiting the most important variables to dominate the trees. The negative aspect of forests that they are less interpretable is not relevant here since we are only interested in predicting the effect of the confounders to X and Y.

Task 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 = i.
    ii. Train a random forest model using randomForest() (with ntree=200 and mtry=2) predicting earnings from confounders on data in folds = i.
    iii. Use models from (i) and (ii) to predict treatment (with type="prob") and outcome for ob- servations 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)?

```
if(file.exists("Task_1_4.rds")){
  residuals <- readRDS("Task_1_4.rds")
} else{

  # a
  ids_folds <- createFolds(d$earnings, k = 5)
  residuals_list <- list()

  # b
  for(i in seq_along(ids_folds)){

    ids_test <- ids_folds[[i]]

    cat("------------------------------\n
        Start Iteration", i, "\nSplitting Data\n")

    testdata <- d[ids_test,] |>
      as.data.frame() |>
      data_select(c("age","education","prior_earnings",
                    "employment_history","urban",
                    "training","earnings"))

    trainingdata <- d[-ids_test,] |>
      as.data.frame() |>
      data_select(c("age","education","prior_earnings",
                    "employment_history","urban",
                    "training","earnings"))

    confounders <- c("age","education","prior_earnings",
                    "employment_history","urban")

    # i
    cat("Calculate RF for X\n")
    m_X <- randomForest(x = trainingdata[,confounders],
                        y = trainingdata$training,
                        ntree = 200, mtry = 2)
    # ii
```

```
    cat("Calculate RF for Y\n")
    m_Y <- randomForest(x = trainingdata[,confounders],
                        y = trainingdata$earnings,
                        ntree = 200, mtry = 2)
    # iii
    cat("Predict Testdata\n")
    X_hat <- predict(m_X, newdata = testdata[,confounders])
    Y_hat <- predict(m_Y, newdata = testdata[,confounders])

    # iv
    cat("Store Residuals\n")
    residuals_list[[i]] <- data.frame(
      fold = i,
      original_index = ids_test,
      X = testdata$training - X_hat,
      Y = testdata$earnings - Y_hat
    )
  }

  residuals <- do.call(rbind, residuals_list)
  residuals <- residuals[order(residuals$original_index), ]

  saveRDS(residuals,"Task_1_4.rds")
}


# c
m_ate <- lm(Y ~ X, residuals)

# f
m_ate |> parameters() |> print_md()
```

Parameter	Coefficient	SE	95% CI	t(11998)	p
(Intercept)	−0.20	0.14	(-0.47, 0.07)	−1.44	0.151
X	12.94	0.35	(12.26, 13.62)	37.37	< .001

```
m_ate |> report_parameters(include_intercept = F)
```

```
  - The effect of X is statistically significant and positive (beta = 12.94, 95% CI [12.26,
13.62], t(11998) = 37.37, p < .001; Std. beta = 0.32, 95% CI [0.31, 0.34])
```

I trust this estimate more then the ones before since random forest prevents overfitting the ATE is not biased towards 0 anymore. Because of that the RF ATE is a bit larger then the decisiontree ATE.

Task 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.

The used models assume all confounding variables are included in the model. Since the true ATE is so different from the estimated ones not all confounding variables were included in the model. This points the researcher towards theorybuilding :)

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)

```
rm(list = ls())

d <- data_read("scholarship.csv")
```

Task 1

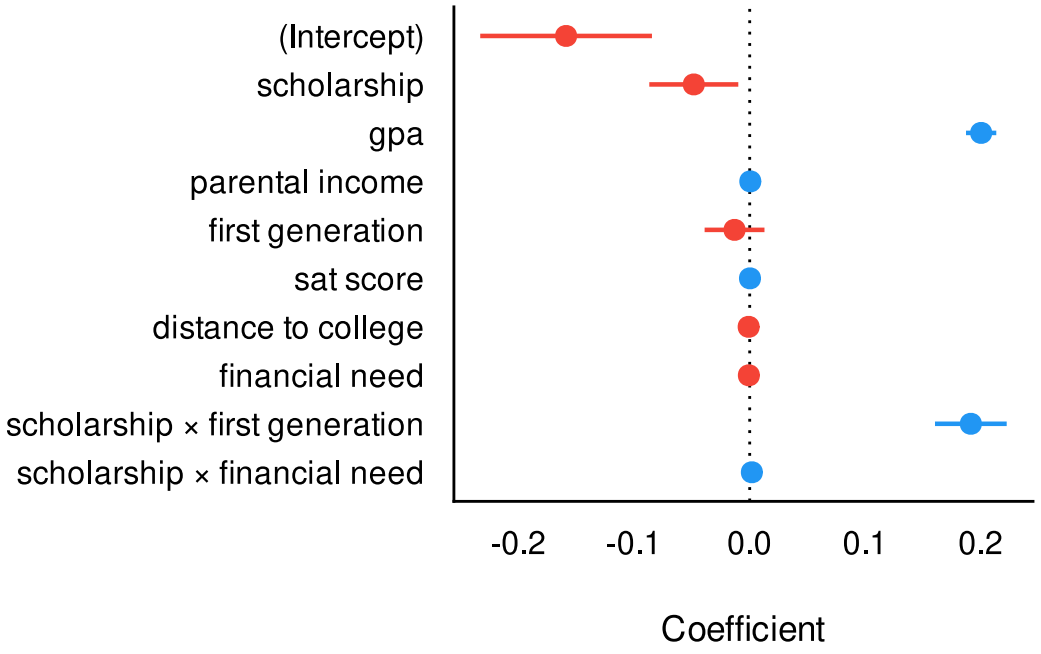
Suppose your co-author, who has done a careful literature review, has found support for two of the variables, first_generation and financial_need, having a moderating effect. What you shall do first is examine whether you find evidence of this in your data. Implement a standard linear regression using 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?

```
m1 <- lm(completed ~ scholarship + gpa + parental_income +
         first_generation + sat_score + distance_to_college + financial_need +
         scholarship:first_generation + scholarship:financial_need,
         data = d)

m1 |> report_table() |> summary() |> print_md()
```

Parameter	Coefficient	95% CI	t(14990)	p	Std. Coef.	Fit
(Intercept)	−0.16	(-0.23, −0.08)	−4.19	< .001	−0.04	
scholarship	−0.05	(-0.09, −9.86e-03)	−2.46	0.014	0.13	
gpa	0.20	(0.19, 0.21)	29.98	< .001	0.23	
parental income	5.28e-04	(3.85e-04, 6.72e-04)	7.22	< .001	0.08	
first generation	−0.01	(-0.04, 0.01)	−0.99	0.321	0.11	
sat score	2.43e-04	(2.01e-04, 2.85e-04)	11.32	< .001	0.09	
distance to college	−9.03e-04	(-1.24e-03, −5.69e-04)	−5.29	< .001	−0.04	
financial need	−7.19e-04	(-1.43e-03, −1.11e-05)	−1.99	0.047	0.02	
scholarship × first generation	0.19	(0.16, 0.22)	12.09	< .001	0.11	
scholarship × financial need	1.87e-03	(1.08e-03, 2.66e-03)	4.65	< .001	0.04	
AICc						14718.71
R2						0.11
R2 (adj.)						0.11

Parameter	Coefficient	95% CI	t(14990)	p	Std. Coef.	Fit
Sigma						0.40

```
m1 |> parameters() |> plot(show_intercept = T)
```



My co-author is partly right: - “first generation” has a statistically significant positive interaction effect - “financial need” has a statistically significant positive interaction effect, but it is so tiny that the significance is probably due to the large observation size. It does not seem to exist.

Task 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?

Research frequently find that the effect of events, exposures, policies (etc) vary across individuals. Usually interactionterms are implemented in the model with the selection of subgroups being based on theory or convention. Limitations of this procedure are that it assumes we know moderators beforehand. While exploratory analysis could help it creates risk of p-hacking and is not feasable for data with too many variables.

Task 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?

Causal trees identify the splits that maximize across-leaf variation in the within-leaf treated–untreated outcome difference. Standard trees just minimize the variance in the leaves. Using trees solves the problem above since they include the most relevant interactions by design.

Task 4

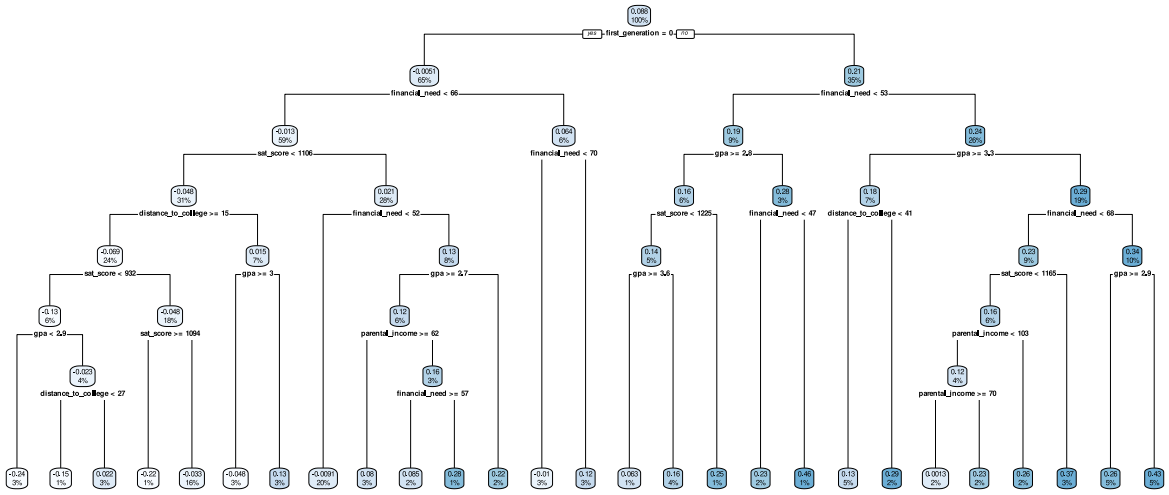
Perform a causal tree analysis by doing the following:

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.
set.seed(5)
if(file.exists("task2_4.rds")){
  m2 <- readRDS("task2_4.rds")
  } else {

    m2 <- causalTree(
      completed ~ gpa + parental_income +
             first_generation + sat_score + distance_to_college + financial_need,
      data      = d,
      treatment = d$scholarship,
      split.Rule = "CT", cv.option = "CT",
      split.Honest = T, cv.Honest = T, split.Bucket = T,
      minsize = 60, cp = 0, bucketNum = 40
    )

    saveRDS(m2, "task2_4.rds")
}

# m2 <- causalTree(
#   completed ~ gpa + parental_income + first_generation +
#     sat_score + distance_to_college + financial_need,
#   data      = d,
#   treatment = d$scholarship,
#   split.Rule = "CT",
#   split.Honest = TRUE,
#   cv.option = "CT",
#   cv.Honest = TRUE,
#   split.Bucket = TRUE,
#   bucketNum = 40,
#   minsize = 60,
#   cp = 0
# )



# b
m2 |> rpart.plot()
```



The population with the biggest treatment effect are students with - no first generation student - have a financial need of at least 68 - have a gpa below 2.9

The population with the smallest treatment effect are students with - the first generation - sat score of less than 932 - distance to college of more than 15 miles - gpa below 2.9

c. We included all variables, so we suppose all variables have an influence on the outcome, but that is not necessarily so.

Task 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.

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)

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?

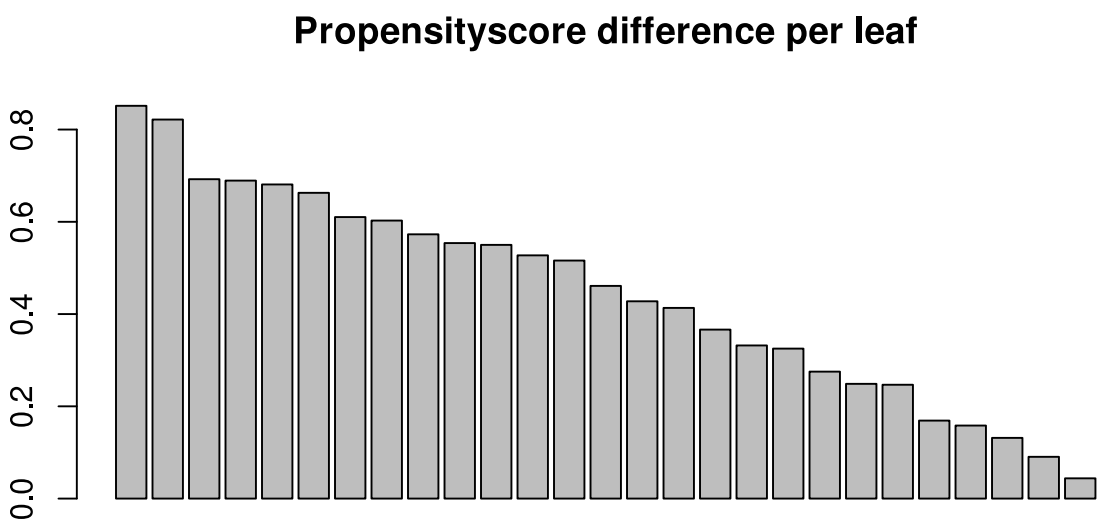
```
# a
m_ps <- glm(scholarship ~ gpa + parental_income +
              first_generation + sat_score +
              distance_to_college + financial_need,
            family = "binomial",
```

```
            data = d)

d$ps_hat <- m_ps |>
  predict(type = "response") |>
  as.numeric()

# b
d <- data.table(d)

d$leaf <- factor(m2$where)

leaf_group_long <- d[, .(
  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
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)]

leaf_balance |> kable()
```

leaf	n_0	n_1	mean_ps_0	mean_ps_1	sd_ps_0	sd_ps_1	SMD_ps
12	130	93	0.4246913	0.5484168	0.1517464	0.1385772	0.8514527
15	282	213	0.3967949	0.5087864	0.1494502	0.1217278	0.8216805
9	95	55	0.3634073	0.4678756	0.1705700	0.1282237	0.6923480
13	1323	1114	0.4194982	0.5240834	0.1630243	0.1395021	0.6893316
16	208	291	0.4900642	0.5933991	0.1698371	0.1311913	0.6809574
34	300	317	0.4617080	0.5514918	0.1534778	0.1146337	0.6628289
18	1766	1265	0.3807935	0.4666484	0.1496207	0.1311412	0.6102646
35	81	91	0.4660410	0.5459750	0.1481456	0.1150429	0.6026797
33	79	58	0.3830011	0.4514854	0.1355346	0.1010592	0.5728684
10	240	177	0.3741601	0.4502100	0.1441455	0.1300950	0.5538954

leaf	n_0	n_1	mean_ps_0	mean_ps_1	sd_ps_0	sd_ps_1	SMD_ps
37	138	171	0.5012876	0.5801155	0.1604781	0.1238293	0.5499755
7	188	201	0.4782521	0.5625178	0.1761478	0.1416117	0.5272689
50	102	333	0.7477117	0.7763115	0.0575133	0.0532564	0.5160032
41	187	513	0.7180357	0.7570764	0.0827440	0.0865649	0.4610599
49	75	204	0.7214137	0.7425716	0.0481313	0.0507776	0.4276732
23	102	197	0.6348092	0.6578473	0.0597068	0.0514613	0.4133377
52	100	643	0.8351172	0.8494616	0.0373100	0.0409250	0.3663080
28	102	394	0.7555206	0.7760850	0.0614917	0.0624030	0.3319583
27	103	299	0.7057624	0.7261344	0.0645151	0.0607458	0.3251249
25	95	262	0.7113184	0.7242799	0.0462857	0.0478958	0.2752050
53	101	703	0.8819694	0.8905410	0.0348651	0.0340553	0.2487225
48	66	235	0.7819908	0.7938036	0.0444596	0.0510528	0.2467677
21	164	235	0.5862516	0.5978796	0.0694551	0.0680902	0.1690700
47	63	222	0.7619663	0.7705220	0.0547601	0.0533246	0.1583001
42	74	210	0.7167703	0.7285805	0.0914241	0.0880601	0.1315789
24	69	100	0.5863733	0.5913258	0.0578105	0.0514313	0.0905161
38	51	120	0.7222593	0.7204272	0.0391284	0.0441869	0.0438970

```
leaf_balance$SMD_ps |>
  barplot(main = "Propensityscore difference per leaf")
```



The in c calculated standardized mean difference of propensity scores indicates how comparable the groups in each leaf are. The propensity score is calculated in the beginning to assess similarity of observations. the `mean_ps_0` and `mean_ps_1` variable are the mean propensity scores for the treatment and control group. Together with the standard deviation of each leaf it is possible to calculate the difference between both groups for each leaf. Therefore leafs with a high balance (small SMD) are well comparable, while leafs with high imbalance (high SMD) differ in their treatment probability.

Only 4 leafs have a propensity score difference lower than 0.2! Most leafs do not allow a sufficient comparison of groups!

Task 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.

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?

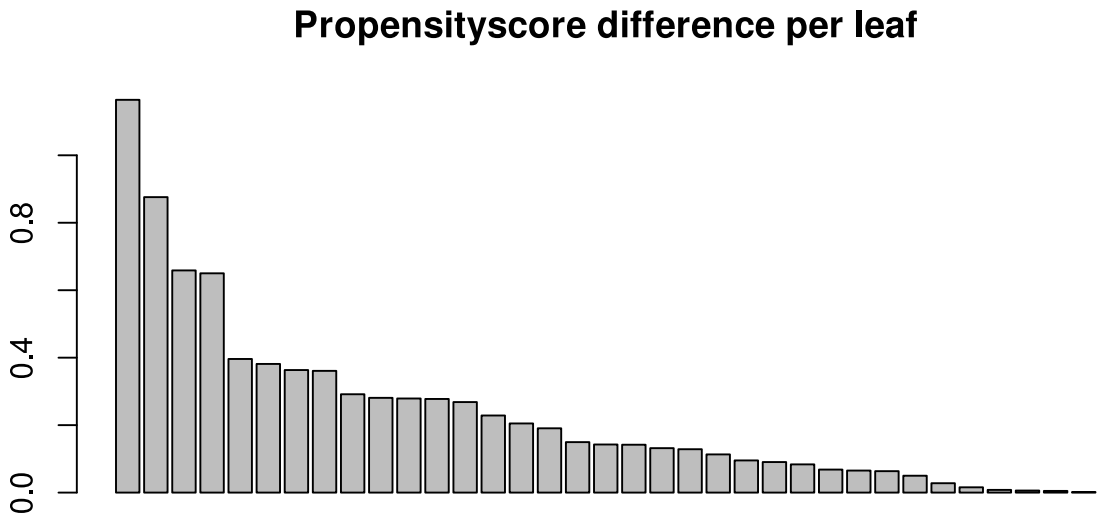
```
# create weights
d$w_ipw <- ifelse(test = d$scholarship == 1,
                  yes = 1/pmax(d$ps_hat, 0.02),
                  no = 1/pmax(1-d$ps_hat, 0.02))


# fit tree
set.seed(5)

if(file.exists("task2_6.rds")){
  m3 <- readRDS("task2_6.rds")
  } else {

    m3 <- causalTree(
      completed ~ gpa + parental_income +
             first_generation + sat_score + distance_to_college + financial_need,
      data      = d,
      treatment = d$scholarship,
      weights = w_ipw,
      split.Rule = "CT", cv.option = "CT",
      split.Honest = T, split.Bucket = T,
      minsize = 60, cp = 0.000043, bucketNum = 40
    )

    saveRDS(m3, "task2_6.rds")
}

# b.
d$leaf_2 <- factor(m3$where)

leaf_group_long_2 <- d[, .(
  n       = .N,
  mean_ps = weighted.mean(ps_hat, w = d$w_ipw),
  sd_ps   = sd(ps_hat)
```

```
  ), by = .(leaf_2, scholarship)]

leaf_group_stats_2 <- data.table::dcast(
  leaf_group_long_2,
  leaf_2 ~ scholarship,
  value.var = c("n", "mean_ps", "sd_ps"),
  fill = NA_real_
  )

leaf_balance_2 <- copy(leaf_group_stats_2)[
  , `:=`(
    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)]

leaf_balance_2 |> kable()
```

leaf_2	n_0	n_1	mean_ps_0	mean_ps_1	sd_ps_0	sd_ps_1	SMD_ps
42	7	99	0.8699643	0.8958324	0.0209583	0.0234043	1.1644421
66	7	60	0.8682397	0.8845810	0.0166790	0.0204420	0.8759453
68	11	62	0.8666732	0.8758524	0.0135497	0.0143105	0.6586974
60	4	60	0.9295203	0.9174807	0.0148176	0.0215969	0.6500805
14	142	30	0.1590812	0.1857832	0.0655638	0.0691967	0.3961448
49	10	21	0.8448095	0.8258344	0.0452084	0.0539042	0.3814340
69	6	61	0.8974588	0.9023201	0.0079475	0.0171724	0.3633227
6	97	30	0.2898870	0.2476222	0.1115567	0.1224176	0.3608885
51	13	50	0.8398264	0.8561758	0.0678681	0.0410238	0.2915576
39	37	113	0.7499938	0.7724096	0.0941780	0.0621931	0.2808841
25	58	78	0.4926034	0.5312090	0.1433936	0.1331911	0.2789695
61	14	95	0.9343623	0.9290803	0.0200424	0.0179582	0.2775802
5	198	244	0.5480125	0.5796634	0.1173914	0.1184817	0.2683696
21	21	53	0.6785176	0.6955420	0.0696733	0.0790171	0.2285408
64	29	177	0.8775664	0.8724971	0.0245640	0.0248944	0.2049888
41	26	173	0.8786166	0.8832178	0.0250768	0.0231676	0.1905960
48	22	83	0.8184012	0.8266501	0.0577688	0.0523734	0.1496073
18	30	100	0.7833728	0.7763922	0.0503774	0.0474831	0.1426037
62	7	48	0.9001213	0.9029265	0.0199499	0.0195841	0.1419084
11	89	138	0.5961494	0.5739408	0.1853769	0.1498824	0.1317504
23	563	411	0.4229967	0.4359843	0.1093817	0.0918858	0.1285729

leaf_2	n_0	n_1	mean_ps_0	mean_ps_1	sd_ps_0	sd_ps_1	SMD_ps
20	180	359	0.6748268	0.6829836	0.0699284	0.0741106	0.1132106
50	128	439	0.7744384	0.7808028	0.0716914	0.0613872	0.0953620
36	195	321	0.6379515	0.6233569	0.1842361	0.1342570	0.0905396
31	233	410	0.6570753	0.6434706	0.1853864	0.1362318	0.0836299
38	61	233	0.7769355	0.7820612	0.0813185	0.0683423	0.0682415
34	380	831	0.6663958	0.6767869	0.1880237	0.1235161	0.0653227
54	95	309	0.7272650	0.7315247	0.0652056	0.0688171	0.0635432
53	126	124	0.4756651	0.4688559	0.1448892	0.1268737	0.0500014
27	1708	1194	0.4114232	0.4075583	0.1475500	0.1299603	0.0277980
24	367	419	0.5252485	0.5235071	0.1130910	0.1093536	0.0156552
59	6	51	0.9256929	0.9258389	0.0193396	0.0171055	0.0079994
12	510	441	0.4707671	0.4717326	0.1693113	0.1489879	0.0060546
8	321	208	0.3840140	0.3833069	0.1397836	0.1446622	0.0049711
28	583	1191	0.6670579	0.6669146	0.0906795	0.0910029	0.0015776

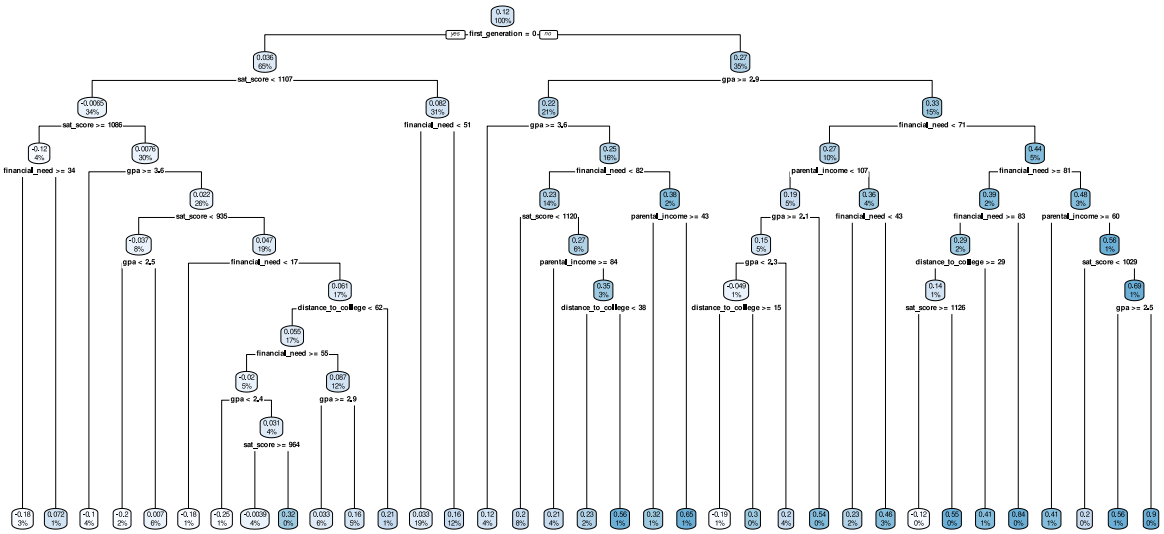
```
leaf_balance_2$SMD_ps |>
  barplot(main = "Propensityscore difference per leaf")
```



The tree is waayyy more balanced now :)

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?

```
m3 |> rpart.plot()
```



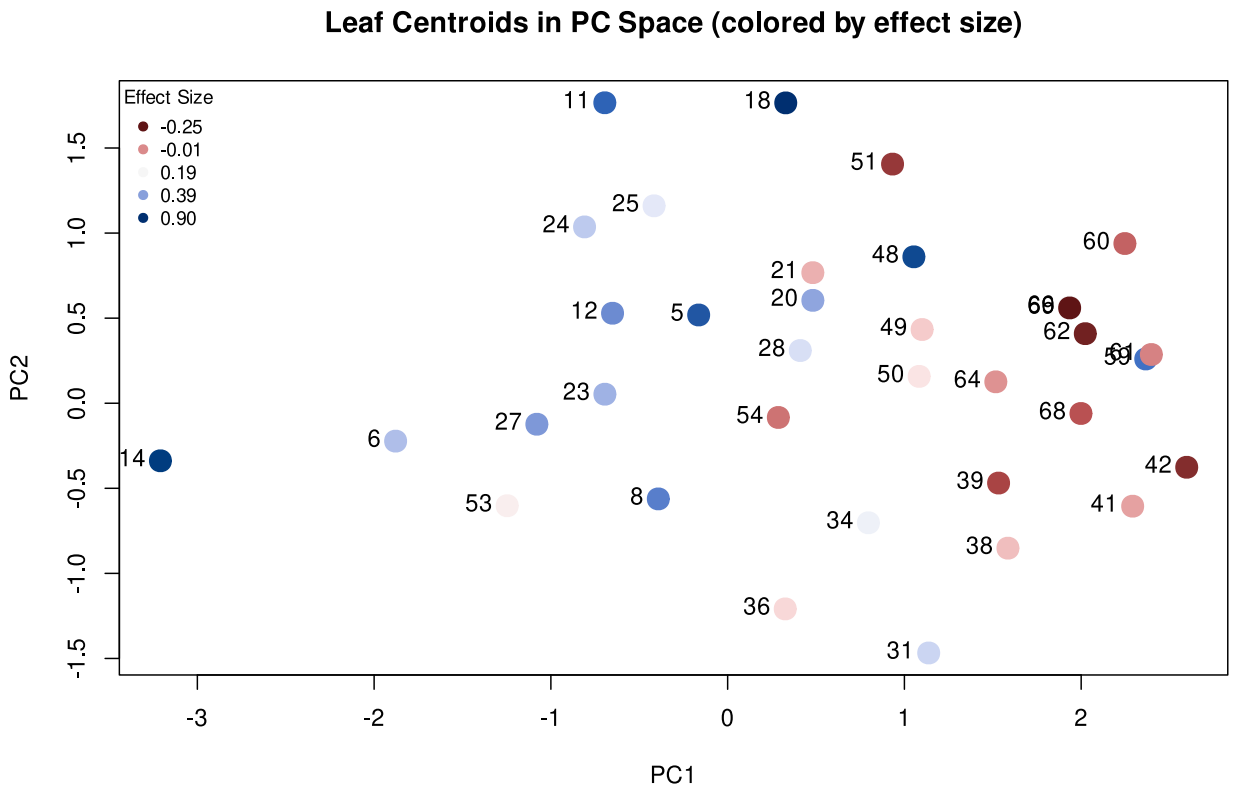
The population with the biggest treatment effect are students with: - first generation student (first_generation = 1) - GPA of at least 2.9 - financial need of at least 71 - parental income of at least 60 - SAT score of at least 1029 - GPA below 2.5

The population with the smallest treatment effect are students with: - no first generation student (first_generation = 0) - SAT score below 1107 - SAT score below 1005 - GPA below 2.6

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).

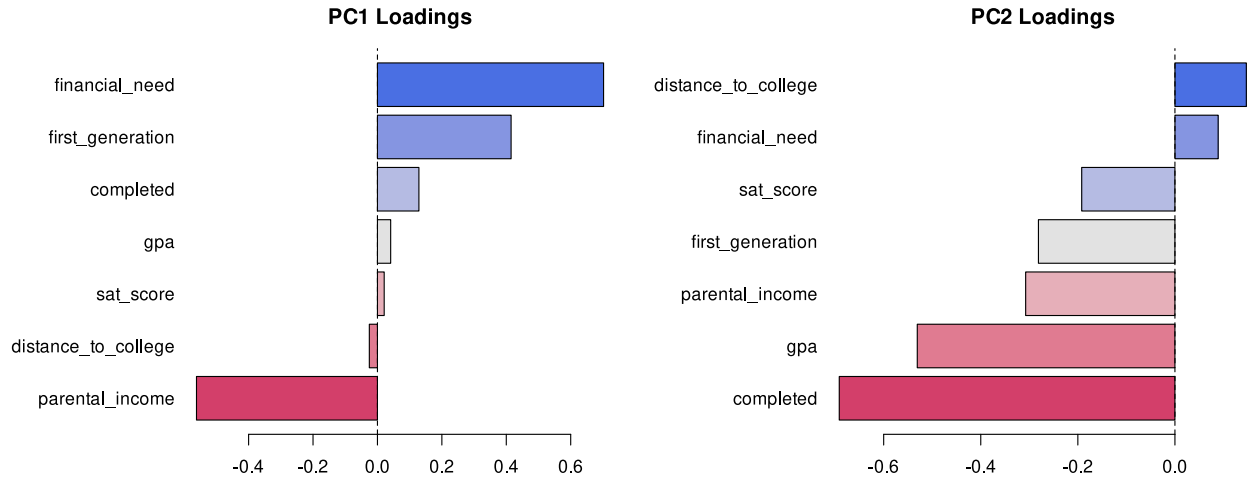
```
leafeffects <- copy(d)[, leaf := m3$where][,
  .(n=.N,
    eff = mean(completed[scholarship==1]) - mean(completed[scholarship==0])),
  by = leaf][order(-eff)]


pca <- d[, c("completed", "gpa", "parental_income", "first_generation",
             "sat_score", "distance_to_college", "financial_need")
         ] |>
  prcomp(scale. = TRUE, center = TRUE)

d$PC1 <- pca$x[, 1]
d$PC2 <- pca$x[, 2]

leaf_summary <- aggregate(
  cbind(PC1, PC2, completed, gpa, parental_income, first_generation,
        sat_score, distance_to_college, financial_need) ~ leaf_2,
  data = d,
  FUN = mean
)

#print(leaf_summary)
```

```r
leaf_pcs <- aggregate(cbind(PC1, PC2) ~ leaf_2, data = d, FUN = mean)

leaf_pcs <- merge(leaf_pcs, leafeffects[, c("leaf", "eff")],
                  by.x = "leaf_2", by.y = "leaf")

plot(leaf_pcs$PC1, leaf_pcs$PC2,
     pch = 19, cex = 2,
     col = hcl.colors(nrow(leaf_pcs), palette = "Blue-Red 3")[rank(leaf_pcs$eff)],
     xlab = "PC1", ylab = "PC2",
     main = "Leaf Centroids in PC Space (colored by effect size)")
text(leaf_pcs$PC1, leaf_pcs$PC2, labels = leaf_pcs$leaf_2, pos = 2)
legend("topleft", pch = 19, title = "Effect Size",
       legend = sprintf("%.2f",
                        quantile(leaf_pcs$eff,
                                 probs = c(0, 0.25, 0.5, 0.75, 1))),
       col = hcl.colors(5, palette = "Blue-Red 3",
                        rev = TRUE),
       cex = 0.8, bty = "n")
```




```r
par(mfrow = c(1, 2), mar = c(4, 10, 3, 1))

# PC1
barplot(sort(pca$rotation[, 1]), horiz = TRUE, las = 1,
        col = hcl.colors(7, palette = "Blue-Red2", rev = T),
        main = "PC1 Loadings")
abline(v = 0, lty = 2)
```

```r
# PC2
barplot(sort(pca$rotation[, 2]), horiz = TRUE, las = 1,
        col = hcl.colors(7, palette = "Blue-Red2", rev = T),
        main = "PC2 Loadings")
abline(v = 0, lty = 2)
```




```r
par(mfrow = c(1, 1))
```

The PCA reveals that many leafs are associated with high PC1 values and most are centered for PC2. PC1 is representing fincancial need where students with rich parents are not in financial need. PC2 is making a distinction between the distance to the college were students who live closer get better grades. We can see that leaf 4 represents the students with rich parents that do not have much better grades than the mean of the students. Most students have above average financial need and no rich parents. Cluster 11 and 18 are students who live far apart and are getting bad grades, while leaf 31 are students that get the best grades. They are also in mediocre financial need.

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?

My coauthor seems to be right! Financial need is the most important divider between groups of students! nice catch!

Part 3

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

Task 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?

The original idea of the IPW was to balance the trees across treated and untreated population. We wanted to make the leafs as comparable as possible. My including probability weighting we limited the information/ influence of less comparable subgroups in the resulting tree. Causal forests algorithms include augmented inverse probability weighting (AIPW) by default. https://chenxing.space/blog/a-walkthrough-of-how-causal-forest-works/#causal-forest

Task 2

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().

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?

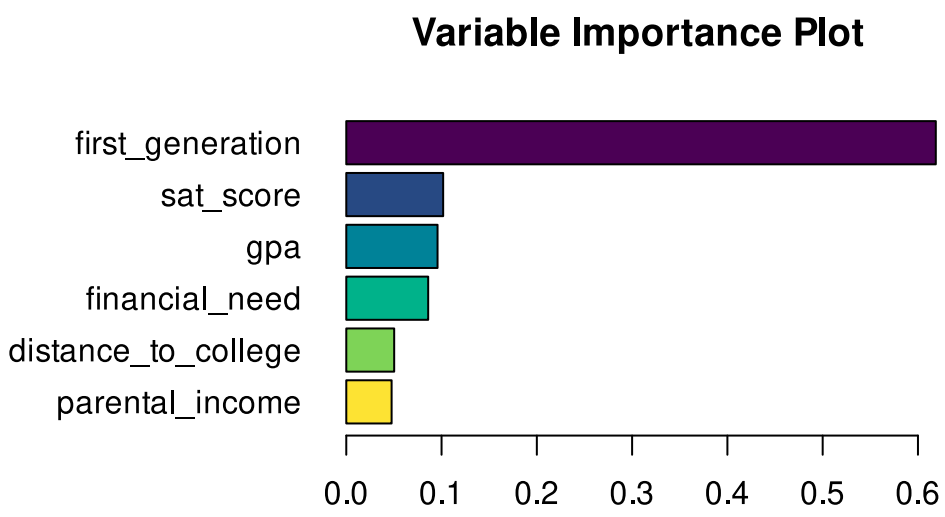
c. For the two variables you identified as most important, please examine how the effects vary along these dimensions. To do so, divide into quintiles of these variables (if continuous) and calculate average treatment effects for each subcategory separately. Plot how the treatment effect varies across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

```
#|label: Task 3.2

d <- data.frame(d)

confounders <- c("gpa", "parental_income", "first_generation",
                 "sat_score", "distance_to_college", "financial_need")

# a

set.seed(5)
if(file.exists("task3_2.rds")){
  m4 <- readRDS("task3_2.rds")
  } else {

    m4 <- causal_forest(d[,confounders] |>
                          as.matrix(),
                        d$completed,
                        d$scholarship,
                        num.trees = 2000,
                        honesty = TRUE)

    saveRDS(m4, "task3_2.rds")
}

average_treatment_effect(m4)
```

```
   estimate     std.err
0.115410857 0.008058812
```

```
# b
par(mar = c(5, 10, 4, 2))

variable_importance(m4) |>
  setNames(confounders) |>
  sort() |>
```

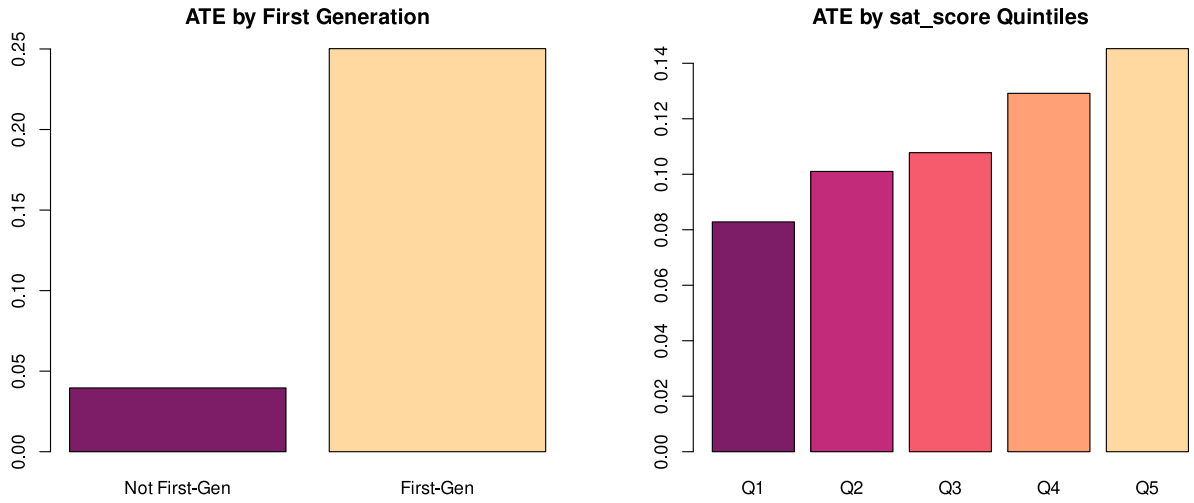
```
  as.table() |>
  barplot(horiz = TRUE,
          las = 1,
          col = hcl.colors(6,rev = T),
          main = "Variable Importance Plot")
```



This result aligns with the causal tree, since the first split in the tree was always `first_generation`.

```
# Get predicted treatment effects
d$tau_hat <- predict(m4)$predictions

# Get top 2 most important variables
var_imp <- variable_importance(m4)
names(var_imp) <- confounders
top_vars <- names(sort(var_imp, decreasing = TRUE)[1:2])


# plot
par(mfrow = c(1, 2),
    mar = c(3, 3, 3, 3))

barplot(tapply(d$tau_hat, d$first_generation, mean),
        names.arg = c("Not First-Gen", "First-Gen"),
        col = hcl.colors(2,palette = "SunsetDark"),
        main = "ATE by First Generation",
        ylab = "ATE")

d$quintile <- cut(d[[top_vars[2]]],
                  breaks = quantile(d[[top_vars[2]]],
                                    probs = seq(0, 1, 0.2)),
                  include.lowest = T,
                  labels = paste0("Q", 1:5))
```

```r
barplot(tapply(d$tau_hat, d$quintile, mean),
        col  = hcl.colors(5,palette = "SunsetDark"),
        main = paste("ATE by", top_vars[2], "Quintiles"),
        ylab = "ATE",
        xlab = "Quintiles")
```



Yes, the causal forest provides important additional insights. While the causal tree identified first-generation status and SAT scores as key moderators through discrete splits, the forest reveals the magnitude and continuous nature of these effects. First-generation students show treatment effects 6 times larger than non-first-generation students (0.25 vs 0.04). SAT scores show a smooth, monotonic increase in treatment effects across quintiles rather than discrete jumps, suggesting scholarships benefit students across the entire SAT distribution with gradually increasing returns. This continuous pattern would be obscured by the tree's binary splits.