

# Lab 6

Thomas Haase

October 14, 2025

## Table of contents

|   |    |
|---|----|
| Part 1: Meta-learners for job training evaluation ..... | 1  |
| Task 1 .....  | 2  |
| Task 2 .....  | 2  |
| Task 3 .....  | 3  |
| Task 4 .....  | 3  |
| Task 5 .....  | 5  |
| Part 2: Heterogeneity I .....                           | 6  |
| Task 1 .....  | 6  |
| Task 2 .....  | 7  |
| Task 3 .....  | 7  |
| Task 4 .....  | 7  |
| Task 5 .....  | 9  |
| Task 6 .....  | 12 |

---

## 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)

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.

```
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:  $X_{\text{tilde}} = X - X_{\text{hat}}$ ,  $Y_{\text{tilde}} = Y - Y_{\text{hat}}$ .

- e. Estimate the ATE by regressing  $Y_{\text{tilde}}$  on  $X_{\text{tilde}}$  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 address 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 observations in fold *i*.
  - iv. Calculate residuals  $X_{\text{tilde}}$  and  $Y_{\text{tilde}}$  for observations in fold *i*.
  - v. Store residuals from fold *i*.
- c. Combine dataset of residuals and regress  $Y_{\text{tilde}}$  on  $X_{\text{tilde}}$  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 than 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 than 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?

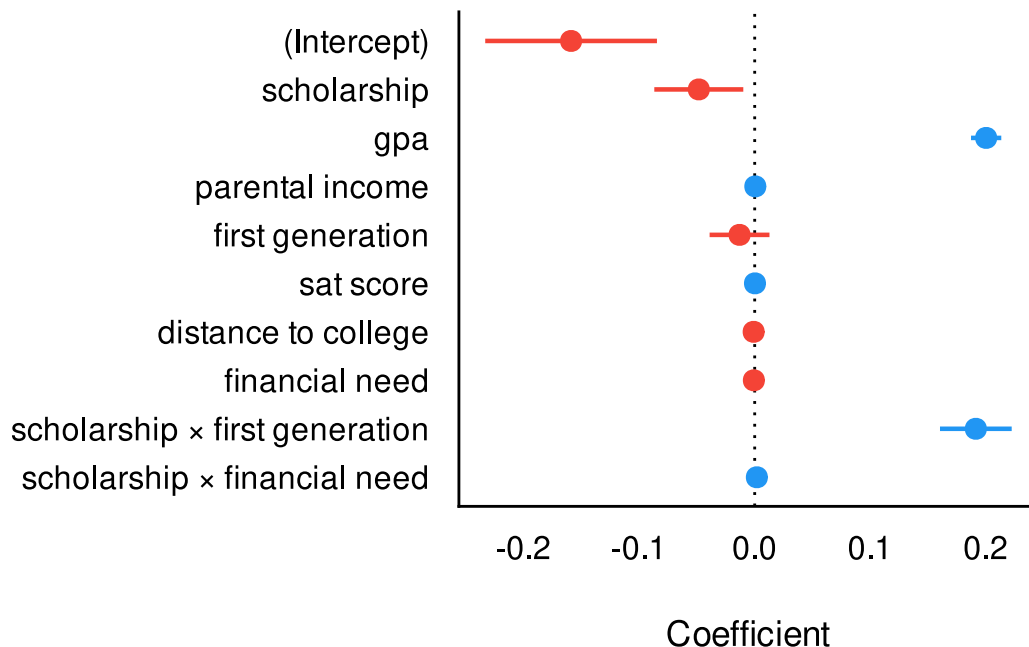
```
m1 <- lm(completed ~ scholarship + gpa + parental_income +
         first_generation + sat_score + distance_to_college + financial_need +
         scholarship:first_generation + scholarship:financial_need,
         data = d)

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 interaction terms 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 feasible 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:

- 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`.
- 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.
- 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?

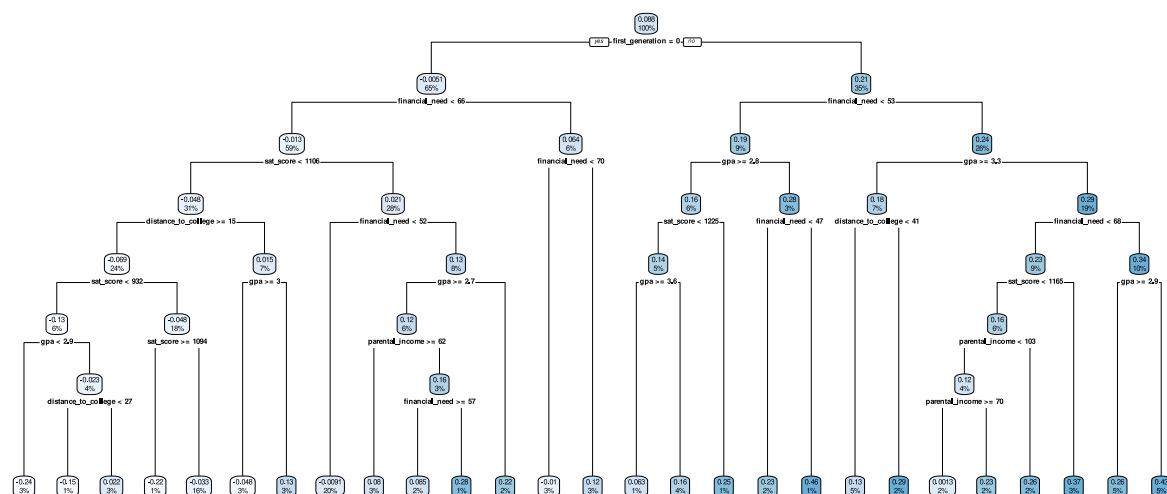
```
# a.
set.seed(5)
if(file.exists("task2_4.rds")){
  m2 <- readRDS("task2_4.rds")
} else {

  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, split.Bucket = T,
    minsize = 60, cp = 0.00001, bucketNum = 40
  )

  saveRDS(m2, "task2_4.rds")
}

# 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)]

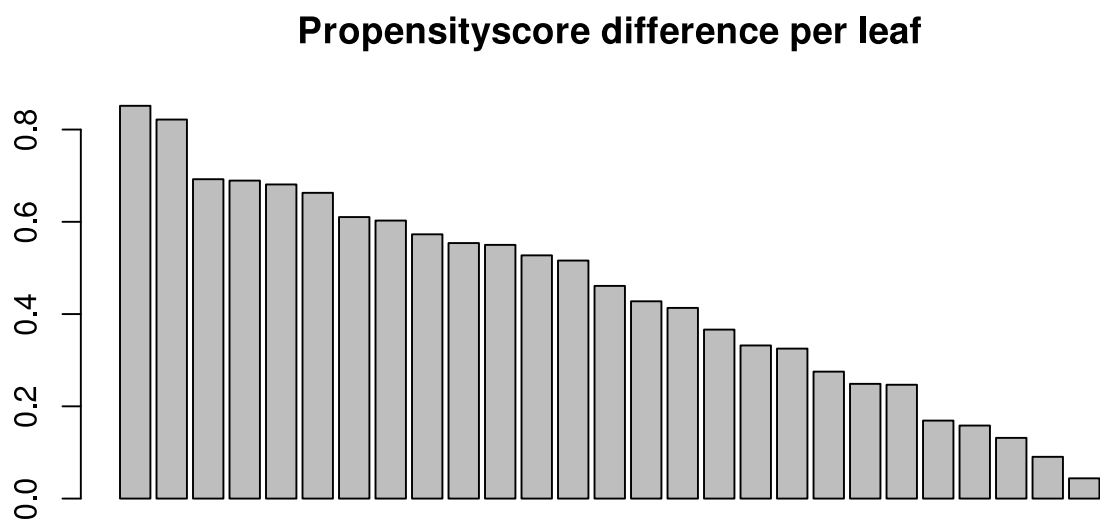
leaf_balance |> kable()

```

| 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:

- 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.
- 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 = mean(ps_hat),
  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()

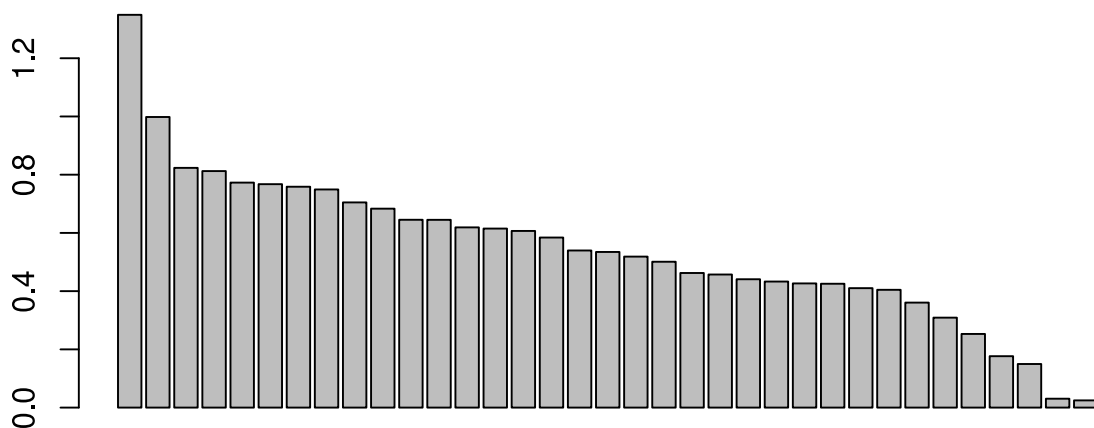
```

| leaf_2 | n_0  | n_1  | mean_ps_0 | mean_ps_1 | sd_ps_0   | sd_ps_1   | SMD_ps    |
|--------|------|------|-----------|-----------|-----------|-----------|-----------|
| 42     | 7    | 99   | 0.8664719 | 0.8964424 | 0.0209583 | 0.0234043 | 1.3491156 |
| 66     | 7    | 60   | 0.8664266 | 0.8850480 | 0.0166790 | 0.0204420 | 0.9981668 |
| 14     | 142  | 30   | 0.1537835 | 0.2092848 | 0.0655638 | 0.0691967 | 0.8234042 |
| 25     | 58   | 78   | 0.4522692 | 0.5646928 | 0.1433936 | 0.1331911 | 0.8123893 |
| 68     | 11   | 62   | 0.8653141 | 0.8760832 | 0.0135497 | 0.0143105 | 0.7728000 |
| 34     | 380  | 831  | 0.5916277 | 0.7137060 | 0.1880237 | 0.1235161 | 0.7674301 |
| 5      | 198  | 244  | 0.5153626 | 0.6048339 | 0.1173914 | 0.1184817 | 0.7586313 |
| 12     | 510  | 441  | 0.4170165 | 0.5365045 | 0.1693113 | 0.1489879 | 0.7492642 |
| 8      | 321  | 208  | 0.3524588 | 0.4527082 | 0.1397836 | 0.1446622 | 0.7047720 |
| 39     | 37   | 113  | 0.7234422 | 0.7779731 | 0.0941780 | 0.0621931 | 0.6833072 |
| 51     | 13   | 50   | 0.8220919 | 0.8582694 | 0.0678681 | 0.0410238 | 0.6451530 |
| 31     | 233  | 410  | 0.5773522 | 0.6822662 | 0.1853864 | 0.1362318 | 0.6449244 |
| 36     | 195  | 321  | 0.5653070 | 0.6650833 | 0.1842361 | 0.1342570 | 0.6189780 |
| 27     | 1708 | 1194 | 0.3751559 | 0.4606484 | 0.1475500 | 0.1299603 | 0.6149055 |
| 11     | 89   | 138  | 0.5280559 | 0.6303363 | 0.1853769 | 0.1498824 | 0.6067659 |
| 53     | 126  | 124  | 0.4398923 | 0.5194358 | 0.1448892 | 0.1268737 | 0.5841067 |
| 23     | 563  | 411  | 0.4029375 | 0.4574485 | 0.1093817 | 0.0918858 | 0.5396424 |
| 21     | 21   | 53   | 0.6643321 | 0.7041516 | 0.0696733 | 0.0790171 | 0.5345483 |
| 60     | 4    | 60   | 0.9275848 | 0.9179798 | 0.0148176 | 0.0215969 | 0.5186249 |
| 38     | 61   | 233  | 0.7510996 | 0.7887285 | 0.0813185 | 0.0683423 | 0.5009762 |
| 24     | 367  | 419  | 0.4989037 | 0.5503342 | 0.1130910 | 0.1093536 | 0.4623471 |

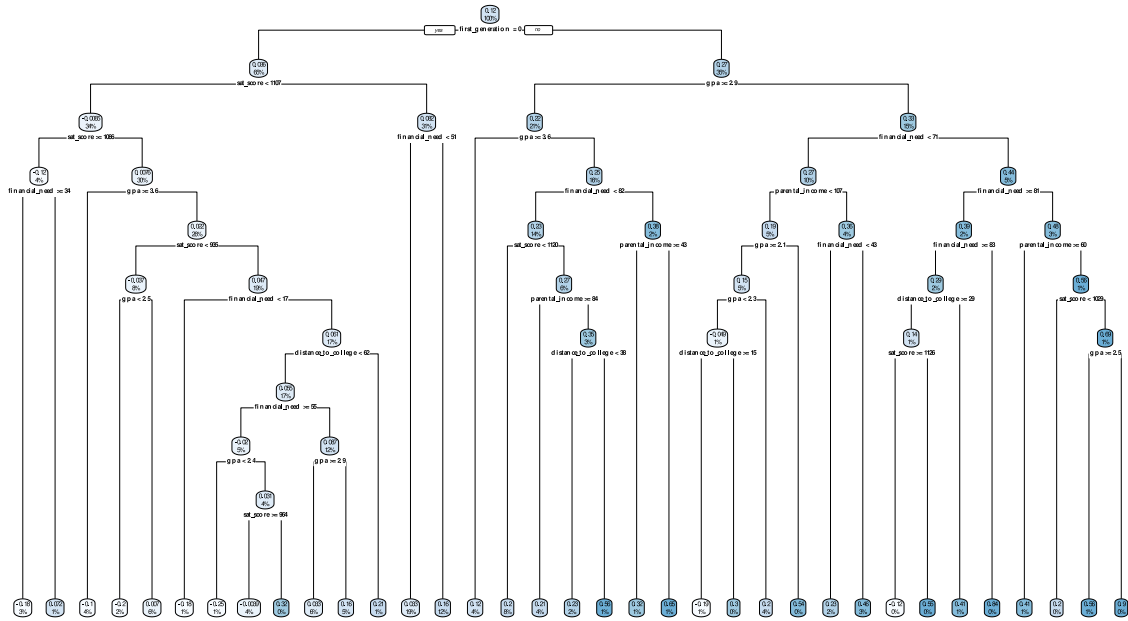
| leaf_2 | n_0 | n_1  | mean_ps_0 | mean_ps_1 | sd_ps_0   | sd_ps_1   | SMD_ps    |
|--------|-----|------|-----------|-----------|-----------|-----------|-----------|
| 48     | 22  | 83   | 0.8051355 | 0.8303386 | 0.0577688 | 0.0523734 | 0.4570981 |
| 20     | 180 | 359  | 0.6591733 | 0.6909363 | 0.0699284 | 0.0741106 | 0.4408485 |
| 50     | 128 | 439  | 0.7573158 | 0.7862110 | 0.0716914 | 0.0613872 | 0.4329617 |
| 28     | 583 | 1191 | 0.6411841 | 0.6799299 | 0.0906795 | 0.0910029 | 0.4265217 |
| 69     | 6   | 61   | 0.8969487 | 0.9026404 | 0.0079475 | 0.0171724 | 0.4253880 |
| 41     | 26  | 173  | 0.8739244 | 0.8838283 | 0.0250768 | 0.0231676 | 0.4102514 |
| 54     | 95  | 309  | 0.7111264 | 0.7382466 | 0.0652056 | 0.0688171 | 0.4045628 |
| 62     | 7   | 48   | 0.8962163 | 0.9033453 | 0.0199499 | 0.0195841 | 0.3606353 |
| 6      | 97  | 30   | 0.2707701 | 0.3069528 | 0.1115567 | 0.1224176 | 0.3089554 |
| 59     | 6   | 51   | 0.9215326 | 0.9261505 | 0.0193396 | 0.0171055 | 0.2529437 |
| 18     | 30  | 100  | 0.7705867 | 0.7792307 | 0.0503774 | 0.0474831 | 0.1765829 |
| 49     | 10  | 21   | 0.8368369 | 0.8293876 | 0.0452084 | 0.0539042 | 0.1497446 |
| 61     | 14  | 95   | 0.9288434 | 0.9294240 | 0.0200424 | 0.0179582 | 0.0305135 |
| 64     | 29  | 177  | 0.8725957 | 0.8732077 | 0.0245640 | 0.0248944 | 0.0247474 |

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

**Propensityscore difference per leaf**



```
rpart.plot(m3)
```



The skewness of the plot is lower, indicating a more stable distribution of leafs, when ignoring the 2 that have a super high propensityscore difference.

- 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()
```

