Lab 5 - R

Table of contents


```
librarian::shelf(
    tidyverse
)
```

```
The 'cran_repo' argument in shelf() was not set, so it will use
cran_repo = 'https://cran.r-project.org' by default.

To avoid this message, set the 'cran_repo' argument to a CRAN
mirror URL (see https://cran.r-project.org/mirrors.html) or set
'quiet = TRUE'.
```

```
Warning: package 'ggplot2' was built under R version 4.3.3
```

1 Data Analysis

1.1 Table Presentation

```
data = read_csv("https://raw.githubusercontent.com/alexanderquispe/CausalAI-Course/main/data/
```

```
Rows: 1739 Columns: 15
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (15): y, w, gender_female, gender_male, gender_transgender, ethnicgrp_as...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
data |> head(3)
```

```
# A tibble: 3 x 15
      y     w gender_female gender_male gender_transgender ethnicgrp_asian
  <dbl> <dbl>         <dbl>       <dbl>              <dbl>           <dbl>
1     1     1             0           1                  0               0
2     0     0             0           1                  0               0
3     0     1             0           1                  0               0
# i 9 more variables: ethnicgrp_black <dbl>, ethnicgrp_mixed_multiple <dbl>,
#   ethnicgrp_other <dbl>, ethnicgrp_white <dbl>, partners1 <dbl>,
#   postlaunch <dbl>, msm <dbl>, age <dbl>, imd_decile <dbl>
```

```
data |> colnames()
```

```
 [1] "y"                      "w"
 [3] "gender_female"          "gender_male"
 [5] "gender_transgender"     "ethnicgrp_asian"
 [7] "ethnicgrp_black"        "ethnicgrp_mixed_multiple"
 [9] "ethnicgrp_other"        "ethnicgrp_white"
[11] "partners1"              "postlaunch"
[13] "msm"                    "age"
[15] "imd_decile"
```

```r
data1 =
    data |> dplyr::select(y, w, contains("gender"), age, imd_decile) |>
    tidyr::pivot_longer(contains("gender"), names_to = "gender") |>
    filter(value > 0) |>
    separate(gender, c("v", "gender")) |>
    mutate(gender = factor(gender, labels = c("male", "female", "transgender"))) |>
    dplyr::select(!c(v, value))
data1 |>
    group_by(w, gender) |>
    summarise(
        n = n(),
        mean_y = mean(y),
        sd_y = sd(y),
        mean_age = mean(age),
        sd_age = sd(age)
    ) |>
    mutate(
        w = ifelse(w == 0, "Control", "Treatment"),
        gender = ifelse(gender == "male", "Male",  ifelse(gender == "female", "Female", "Tra
    ) |>
    rename(
        Group = 1,
        Gender = 2,
        "Mean - Y" = 4,
        "sd - Y" = 5,
        "Mean - Age" = 6,
        "sd - Age" = 7
    ) |> knitr::kable(caption = "Descriptive Statisctics by Group and Gender")
```

```
`summarise()` has grouped output by 'w'. You can override using the `.groups`
argument.
```

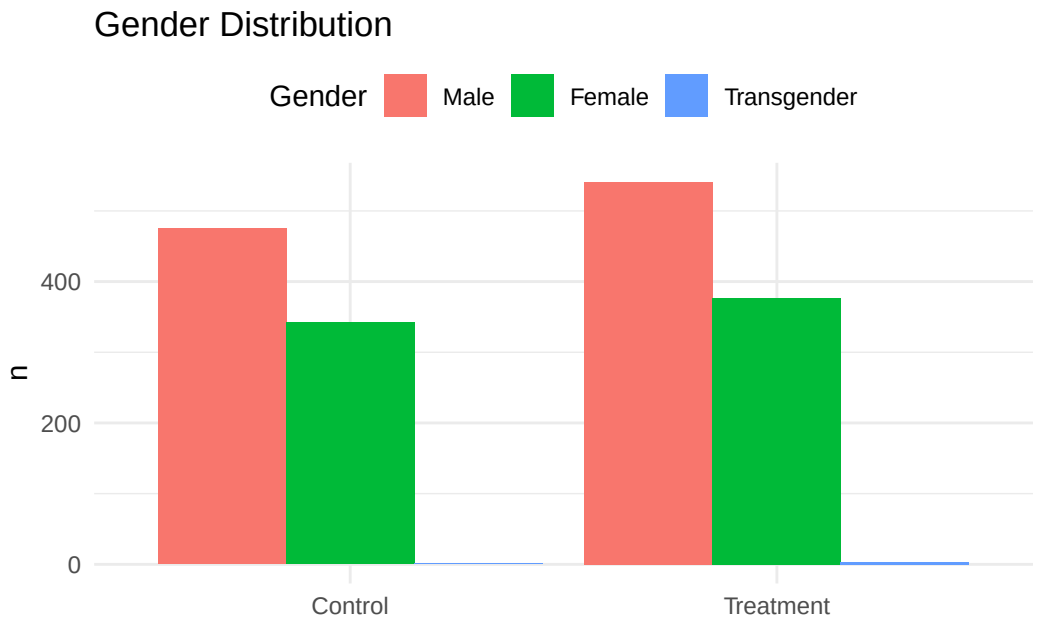
Table 1: Descriptive Statisctics by Group and Gender

Group	Gender	n	Mean - Y	sd - Y	Mean - Age	sd - Age
Control	Male	475	0.2084211	0.4066076	22.74105	3.591231
Control	Female	342	0.2163743	0.4123757	23.49123	3.547530
Control	Transgender	1	0.0000000	NA	17.00000	NA
Treatment	Male	541	0.5341959	0.4992909	22.92052	3.500154
Treatment	Female	377	0.3899204	0.4883801	23.50663	3.575267

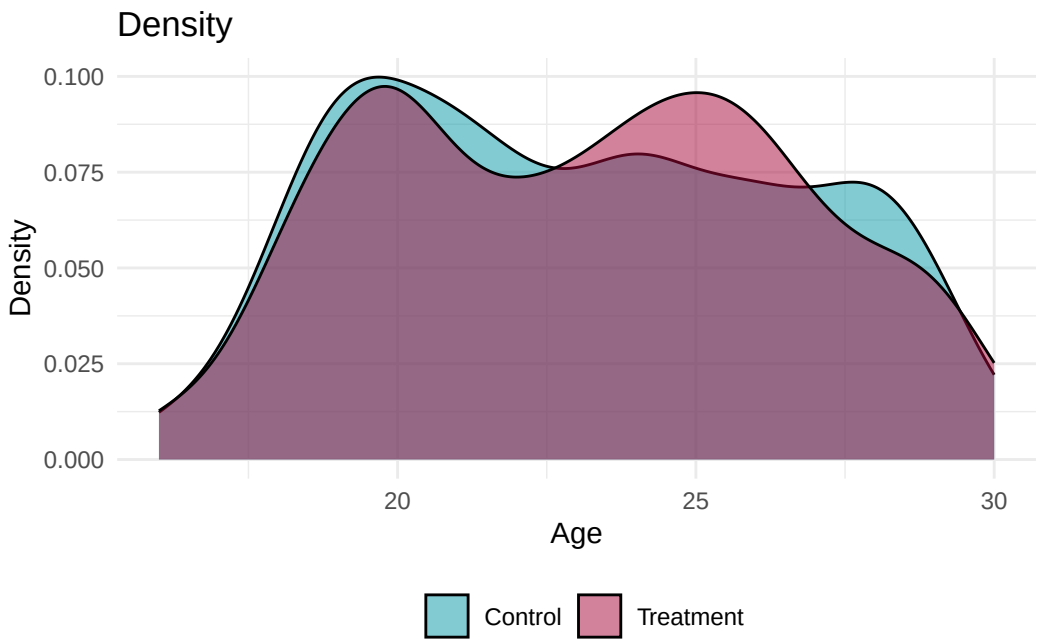
Group	Gender	n	Mean - Y	sd - Y	Mean - Age	sd - Age
Treatment	Transgender	3	1.0000000	0.0000000	22.33333	3.214550

1.2 Graphs - Final output

```
theme_set(theme_minimal())
data1 |>
    ggplot() +
    aes(factor(w, labels = c("Control", "Treatment")), fill = factor(gender, labels = c("Mal
    geom_bar( position = "dodge") +
    labs(
        x = "",
        y = "n",
        title = "Gender Distribution",
        fill = "Gender"
    ) +
    theme(
        legend.position = "top"
    )
```



```r
data1 |>
    ggplot() +
    aes(age,
    fill = factor(w, labels = c("Control", "Treatment"))
    ) +
    geom_density(alpha = .5) +
    scale_fill_manual(values = c("#0798a8", "#a8073a")) +
    labs(
        x = "Age", y = "Density",
        title = "Density",
        fill = ""
    ) +
    theme(
        legend.position = "bottom"
    )
```



2 Linear Regression Analysis

2.1 LM 1

```
lm1 =lm(y ~ w, data = data) |>
    broom::tidy()
```

2.2 LM 2

```
base_formula = y ~ w + age + imd_decile
lm2 = lm(base_formula, data = data1) |>
    broom::tidy()
```

2.3 Double Lasso

```
cnt <- c(
    "w",
    "gender_female",
    "gender_male",
    "age",
    "imd_decile"
    )

formula <- as.formula(
    paste(
        "y ~ ",  paste(cnt, collapse = " + ")
        )
)


X <- model.matrix(formula, data = data)[,-1]
Y <- data$y


coefs_glm =
    glmnet::cv.glmnet(X, Y, alpha = .1) |>
    coef(s = "lambda.min") |>
```

```
    as.matrix() |>
    as_tibble(rownames='var') |>
    filter(s1 != 0) |>
    filter(!stringr::str_detect(var, "ntercept")) |>
    pull(var)

lm3 = lm(
    as.formula(
        paste(
            "y ~ ",
            paste(coefs_glm, collapse = "+")
        )
    ),
    data = data
) |> broom::tidy()
```

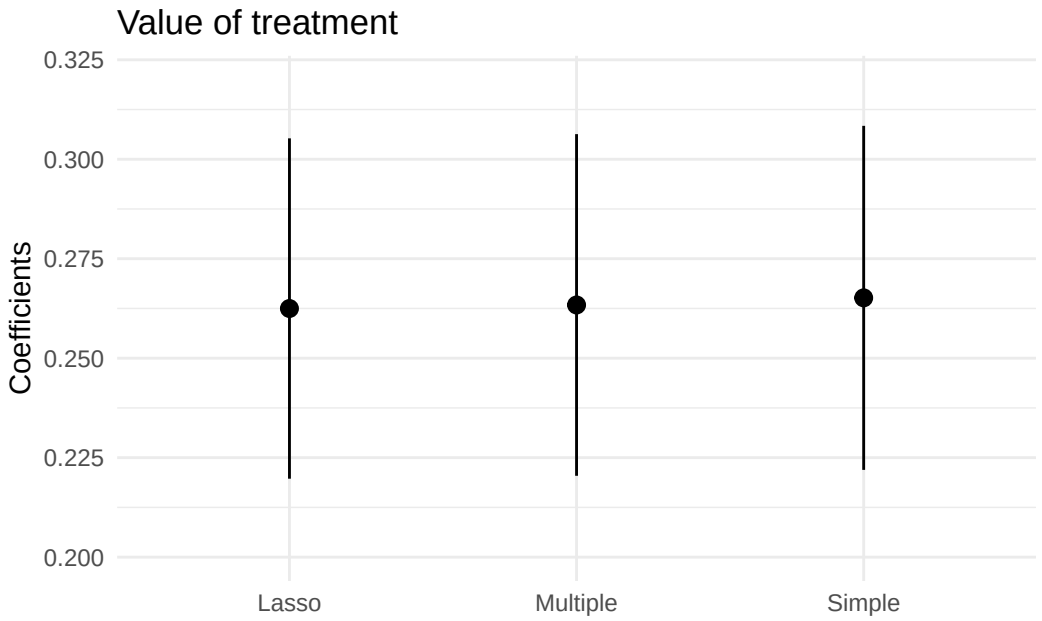
2.4 Results

```
table = bind_rows(
    lm1 |> mutate(model = "Simple"),
    lm2 |> mutate(model = "Multiple"),
    lm3 |> mutate(model = "Lasso")
    ) |>
    filter(term == "w")
table |> knitr::kable()
```

term	estimate	std.error	statistic	p.value	model
w	0.2651644	0.0220586	12.02088	0	Simple
w	0.2633769	0.0219073	12.02233	0	Multiple
w	0.2624866	0.0218226	12.02821	0	Lasso

```
table |>
    ggplot() +
    aes(
        model, estimate,
        ymin = estimate - 1.96*std.error,
        ymax = estimate + 1.96*std.error
    ) +
```

```
    geom_point() +
    geom_pointrange() +
    ylim(c(0.2, 0.32)) +
    labs(
        x = "",
        y = "Coefficients",
        title = "Value of treatment"
    )
```



3 Non Linear Methods

```
splits = rsample::initial_split(data, strata = y)

data_test = rsample::testing(splits)
data_train = rsample::training(splits)

y_train = data_train |> select(y)
T_train = data_train |> select(w)
x_train = data_train |> select(!c(y, w))
```

```r
y_test = data_test |> select(y)
T_test = data_test |> select(w)
x_test = data_test |> select(!c(y, w))
```

```r
lm_yd = function(y, d, md){
    df =tibble(y = unlist(y) , d = unlist(d) )
    # print(head(df))
    lm(y ~ d, data = df) |>
        broom::tidy() |>
        mutate(
            cols = c("I", "d"),
            model = md
        )
}
```

3.1 Lasso

```r
lasso_y <- glmnet::cv.glmnet(as.matrix(x_train), y_train |> as.matrix(), alpha = 1, nfolds =
lasso_t <- glmnet::cv.glmnet(as.matrix(x_train), T_train |> as.matrix(), alpha = 1, nfolds =

y_train_hat <- predict(lasso_y, newx = as.matrix(x_test))
T_train_hat <- predict(lasso_t, newx = as.matrix(x_test))

y_r1 <- y_train - y_train_hat
T_r1 <- T_train - T_train_hat

l1 = lm_yd(y_r1, T_r1, "Lasso")
```

3.2 Regression Trees

```r
tree_y <- rpart::rpart(y ~ . - w, data = data_train, method = "anova")
tree_t <- rpart::rpart(w ~ . - y, data = data_train, method = "anova")

Y_predicted <- predict(tree_y, newdata = data_test)
T_predicted <- predict(tree_t, newdata = data_test)
y_r2 <- y_test - Y_predicted
T_r2 <- T_test - T_predicted
```

```
l2 = lm_yd(y_r2, T_r2, "Reg. Trees")
```

3.3 Boosting Trees

```
boost_y <- gbm::gbm(y ~ . - w, data = data_train, distribution='bernoulli')
boost_t <- gbm::gbm(w ~ . - y, data = data_train, distribution='bernoulli')


Y_predicted <- predict(boost_y, newdata = data_test, n.trees = 100)
T_predicted <- predict(boost_t, newdata = data_test, n.trees = 100)
y_r3 <- y_test - Y_predicted
T_r3 <- T_test - T_predicted

l3 = lm_yd(y_r3, T_r3, "Bost. Trees")
```

3.4 Regresssion Forest

```
model_y <- randomForest::randomForest(y ~ . - w, data = data_train)
```

```
Warning in randomForest.default(m, y, ...): The response has five or fewer
unique values.  Are you sure you want to do regression?
```

```
model_t <- randomForest::randomForest(w ~ . - y, data = data_train)
```

```
Warning in randomForest.default(m, y, ...): The response has five or fewer
unique values.  Are you sure you want to do regression?
```

```
Y_predicted <- predict(model_y, newdata = data_test)
T_predicted <- predict(model_t, newdata = data_test)
y_r4 <- y_test - Y_predicted
T_r4 <- T_test - T_predicted

l4 = lm_yd(y_r4, T_r4, "Reg. Forest")
```

3.5 Results

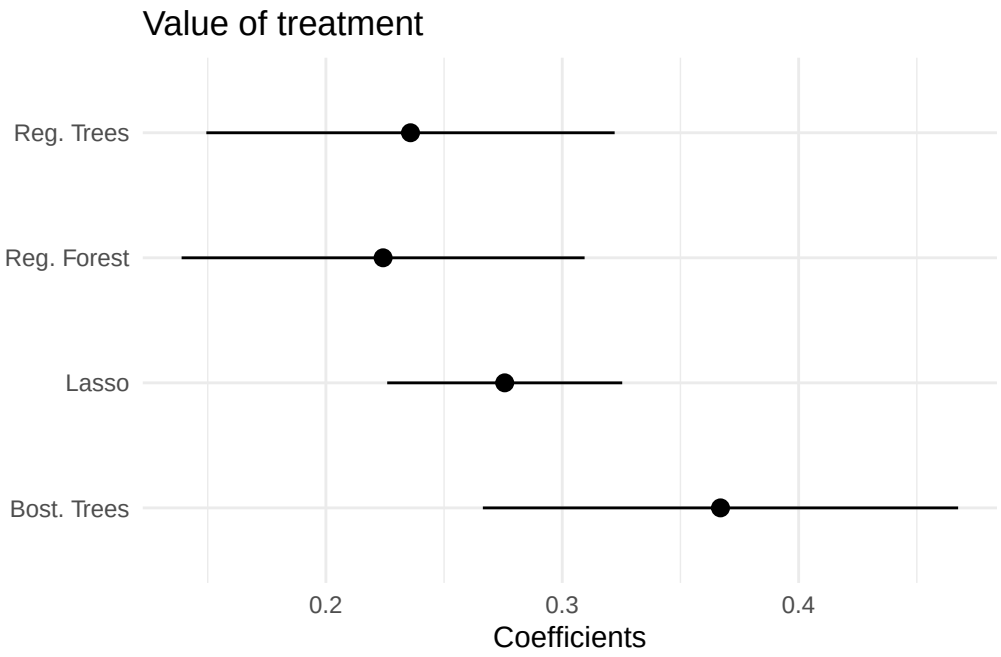
```
results = bind_rows(
    l1, l2, l3, l4
) |>
    filter(term == "d") |>
    select(!cols) |>
    arrange(estimate)

results |>
    knitr::kable(caption="Regression values")
```

Table 3: Regression values

term	estimate	std.error	statistic	p.value	model
d	0.2242041	0.0434896	5.155352	4e-07	Reg. Forest
d	0.2357984	0.0440641	5.351257	1e-07	Reg. Trees
d	0.2756558	0.0253648	10.867670	0e+00	Lasso
d	0.3669385	0.0512926	7.153827	0e+00	Bost. Trees

3.5.1 Plot

```
results |>
    ggplot() +
    aes(
        model, estimate,
        ymin = estimate - 1.96*std.error,
        ymax = estimate + 1.96*std.error
    ) +
    geom_point() +
    geom_pointrange() +
    labs(
        x = "",
        y = "Coefficients",
        title = "Value of treatment"
    ) +
    coord_flip()
```


Value of treatment
Reg. Trees
Reg. Forest
Lasso
Bost. Trees
0.2
0.3
0.4
Coefficients