hw5

1

a

Identify the ATE under conditional ignorability:

$$
\begin{aligned}
E[Y(1) - Y(0)] &= E\Big[E[Y(1) \mid X]\Big] - E\Big[E[Y(0) \mid X]\Big] && \text{iterated expectation} \\
&= E\Big[E[Y(1) \mid X,\ D = 1]\Big] - E\Big[E[Y(0) \mid X,\ D = 0]\Big] && \text{CI}
\end{aligned}
$$

b

We can write that

$$
\begin{aligned}
\hat{\tau}_{Lin} &= \underset{\tau}{\operatorname{argmin}} \sum_{i=1}^{n} \left[Y_i - \left(\tau D_i + (X_i - \bar{X}_1)^T \hat{\beta}_{Lin} + D_i (X_i - \bar{X}_1)^T \hat{\gamma}_{Lin} \right) \right]^2 \\
&= \underset{\tau}{\operatorname{argmin}} \sum_{i=1}^{n} \left[\left(Y_i - (X_i - \bar{X}_1)^T \hat{\beta}_{Lin} - D_i (X_i - \bar{X}_1)^T \hat{\gamma}_{Lin} \right) - \tau D_i \right]^2 \\
&= \frac{1}{n_1} \sum_{D_i=1} \left(Y_i - (X_i - \bar{X}_1)^T \hat{\beta}_{Lin} - (X_i - \bar{X}_1)^T \hat{\gamma}_{Lin} \right) - \frac{1}{n_0} \sum_{D_i=0} \left(Y_i - (X_i - \bar{X}_1)^T \hat{\beta}_{Lin} \right) \\
&= \underbrace{\left[\frac{1}{n_1} \sum_{D_i=1} Y_i - \frac{1}{n_0} \sum_{D_i=0} Y_i \right]}_{\hat{\tau}_{dim}} - \frac{1}{n_1} \sum_{D_i=1} (X_i - \bar{X}_1)^T \hat{\beta}_{Lin} - \frac{1}{n_1} \sum_{D_i=1} (X_i - \bar{X}_1)^T \hat{\gamma}_{Lin} + \\
&\quad \frac{1}{n_0} \sum_{D_i=0} (X_i - \bar{X}_1)^T \hat{\beta}_{Lin} \\
&= \hat{\tau}_{dim} - \frac{1}{n_1} \sum_{D_i=1} X_i^T \hat{\beta}_{Lin} + \bar{X}_1{}^T \hat{\beta}_{Lin} - \frac{1}{n_1} \sum_{D_i=1} X_i^T \hat{\gamma}_{Lin} + \bar{X}_1{}^T \hat{\gamma}_{Lin} + \frac{1}{n_0} \sum_{D_i=0} X_i^T \hat{\beta}_{Lin} - \bar{X}_1{}^T \hat{\beta}_{Lin} \\
&= \hat{\tau}_{dim} - \frac{1}{n_0} \sum_{D_i=0} X_i^T \hat{\beta}_{Lin} - \bar{X}_1{}^T \hat{\beta}_{Lin} \\
&= \frac{1}{n_1} \sum_{D_i=1} Y_i - \frac{1}{n_0} \sum_{D_i=0} Y_i - \frac{1}{n_0} \sum_{D_i=0} X_i^T \hat{\beta}_{Lin} - \frac{1}{n_1} \sum_{D_i=1} X_i^T \hat{\beta}_{Lin}
\end{aligned}
$$

but now,

$$
\frac{1}{n_0} \sum_{D_i=0} Y_i - \frac{1}{n_0} \sum_{D_i=0} X_i^T \hat{\beta}_{Lin} = \frac{1}{n_0} \sum_{D_i=0} \left(Y_i - X_i^T \hat{\beta}_{Lin} \right) = 0
$$

Because the residuals must sum to zero for this portion of the model. And so we are left with

$$
\frac{1}{n_1} \sum_{D_i=1} Y_i - \frac{1}{n_1} \sum_{D_i=1} X_i^T \hat{\beta}_{Lin}
$$

as desired.

2

a

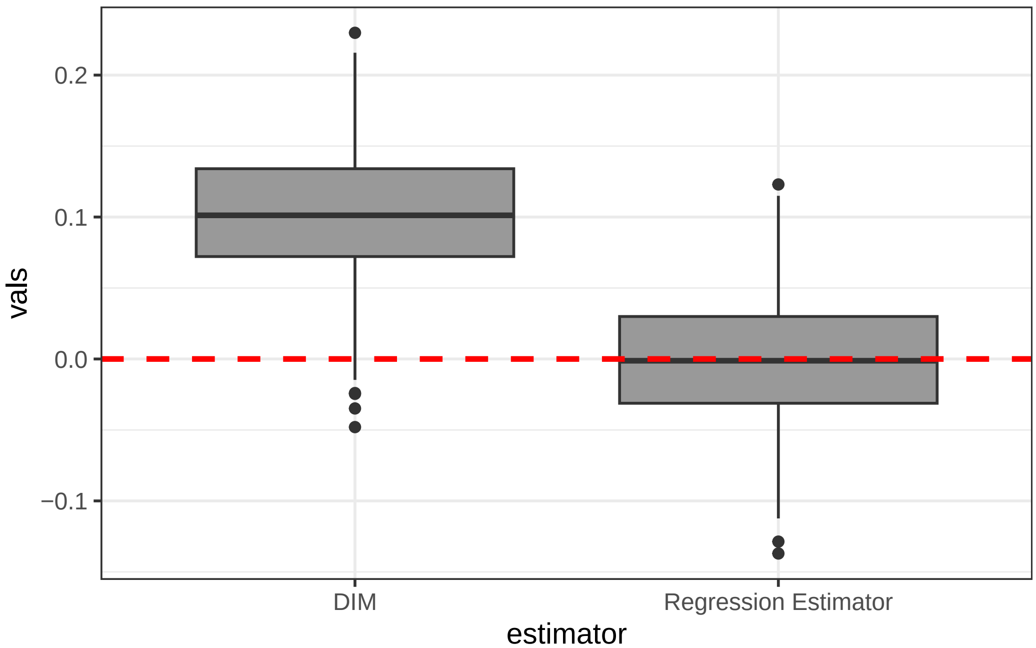
```r
library(tidyverse)
dgp_p2 <- function(n) {
  tibble(
    x1 = rnorm(n, 0, 1),
    x2 = rchisq(n, 1),
    prob_D = exp(0.5*x1 + 0.5*x2 - 0.5)/(1 + exp(0.5*x1 + 0.5*x2 - 0.5)),
    D = rbinom(n, 1, prob_D),
    prob_Y = exp(0.6*x1 + 0.2*x2 + 0.5*x1*x2)/(1 + exp(0.6*x1 + 0.2*x2 + 0.5*x1*x2)),
    Y = rbinom(n, 1, prob_Y)
  ) %>%
    select(-prob_D, - prob_Y)
}
```

```r
att_est <- function(dat) {
  mod_y <- glm(Y ~ x1 + x2 + x1*x2, dat[dat$D == 0, ], family = "binomial")
  p1 <- mean(dat[dat$D == 1, "Y", drop = T])
  p2 <- mean(predict(mod_y, newdata = dat[dat$D==1, ], type = "response"))
  est_att <- p1 - p2

  return(est_att)
}

bootstrap <- function(dat){
  ests <- rep(0, 500)
  for(i in 1:500) {
    dat_boot <- dat %>%
      slice_sample(n = 500, replace = T)

    ests[i] <- att_est(dat_boot)
  }
  return(ests)
}


sim <- function(df) {
  dim <- lm(Y ~ D, df)$coefficients[2]
  est_att <- att_est(df)
```

```
  boots <- bootstrap(df)
  ci <- quantile(boots, c(0.025, 0.975))

  tibble(
    dim = dim,
    att_hat = est_att,
    lower = ci[1],
    upper = ci[2]
  )
}

data_list <- list()
for(i in 1:500){
  data_list[[i]] <- dgp_p2(500)
}

res <- data_list %>%
  map_dfr(sim)
```

Looks unbiased

```
library(here)

res <- read_csv(here("data", "hw5_res.csv"))

res %>%
  pivot_longer(cols = c(dim, att_hat), names_to = "estimator", values_to = "vals") %>%
   mutate(estimator = case_when(
    estimator == "att_hat" ~ "Regression Estimator",
    estimator == "dim" ~ "DIM"
  )) %>%
  ggplot(aes(x = estimator, y = vals)) +
  geom_boxplot(fill = "grey60") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red", linewidth = 1) +
  theme_bw()
```



Correct coverage

```
res %>%
  rowwise() %>%
  mutate(covers = between(0, lower, upper)) %>%
  ungroup() %>%
  summarise(coverage = mean(covers))
```

```
# A tibble: 1 x 1
  coverage
     <dbl>
1    0.944
```

3

a

non random assignment and clustered data

```
dgp_p3 <- function(n){
```

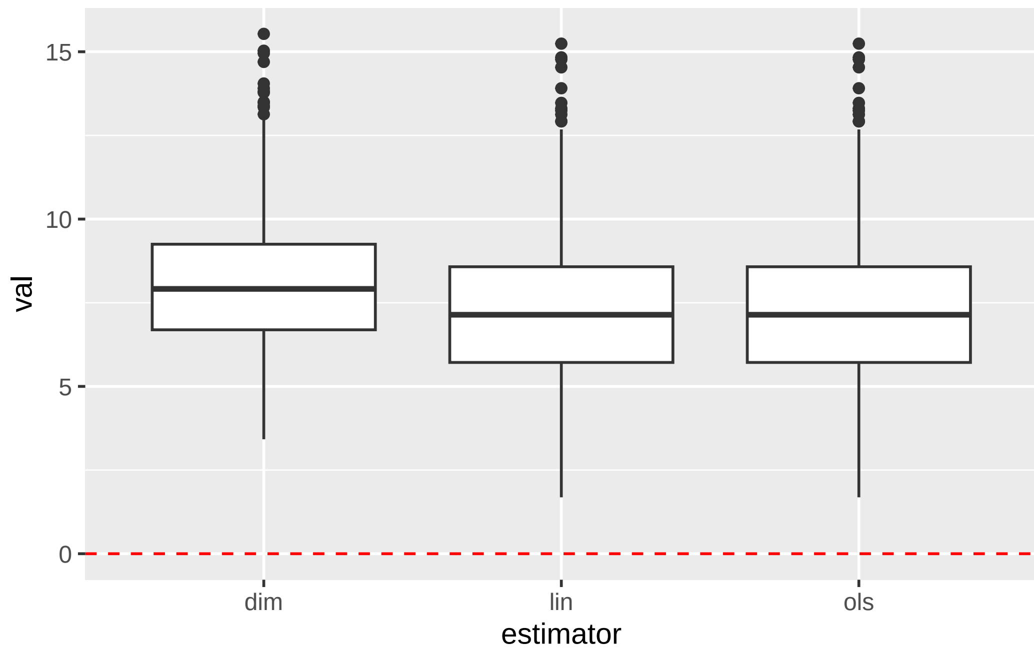
```r
  # grouped data
  X <- rnorm(n)
  group <- rep(seq(1, 10), n/10)
  group_intercepts <- rnorm(10, 0, 5)
  Y <- 10 + 3*X +  group_intercepts[group]

  ## assignments
  prob <- exp(X + group_intercepts[group])/(1 + exp(X + group_intercepts[group]))
  D <- rbinom(n, 1, prob)

  tibble(
    x = X,
    d = D,
    y = Y
  )
}
```

```r
sim <- function(data){
  data$x1_center <- data$x - mean(data[data$d == 1, "x", drop = T])

  # dim
  dim <- coef(lm(y ~ d, data))["d"]

  # lin
  lin <- coef(lm(y ~ d + x1_center + d*x1_center, data))["d"]

  # ols
  y1 <- mean(data[data$d == 1, "y", drop = T])
  ols_mod <- lm(y ~ x, data[data$d == 0, ])
  ols_pred <- mean(predict(ols_mod, newdata = data[data$d == 1, ]))
  ols <- y1 - ols_pred

  tibble(
    estimator = c("dim", "ols", "lin"),
    val = c(dim, ols, lin)
  )
}
```

```r
sim_data <- list()
for(i in 1:1000) {
  sim_data[[i]] <- dgp_p3(1000)
```

```
}

res <- sim_data %>%
  map_dfr(sim)
```

```
res %>%
  ggplot(aes(x = estimator, y = val)) +
  geom_boxplot() +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed")
```



b

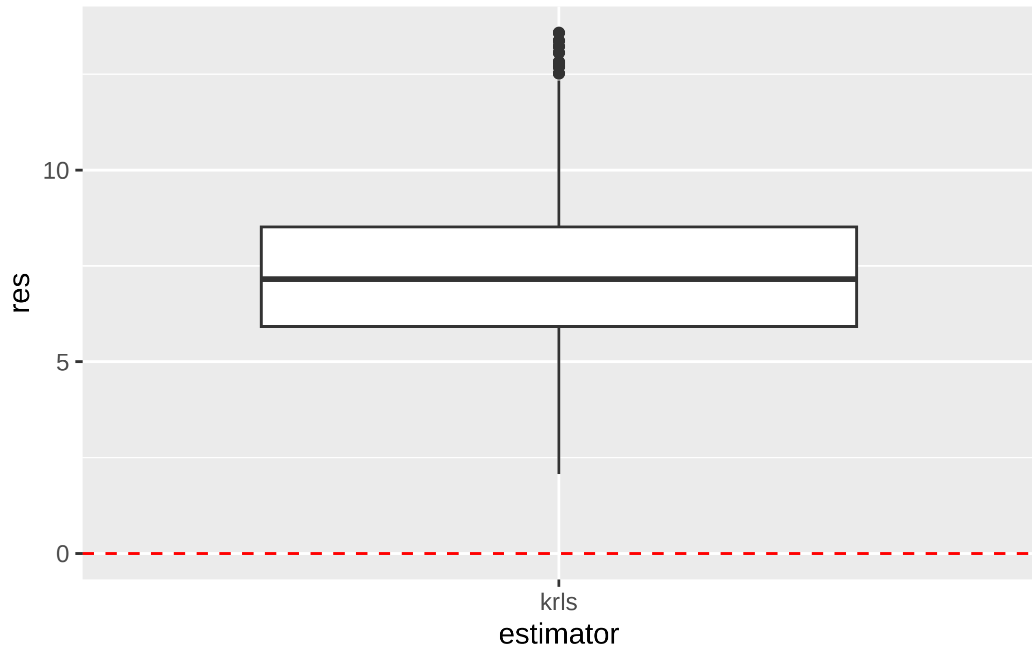
```
library(KRLS)
```

```
## KRLS Package for Kernel-based Regularized Least Squares.

## See Hainmueller and Hazlett (2014) for details.
```

```r
krls_sim <- function(data) {

  data_train <- data %>%
    filter(d == 0)

  data_pred <- data %>%
    filter(d == 1)

  X <- as.matrix(data_train$x)
  y <- data_train$y

  mod <- krls(
    X = X, y = y,
    whichkernel = "gaussian", sigma = ncol(X),
    derivative = F, vcov = F
  )

  preds <- predict(mod, newdata = as.matrix(data_pred$x))

  return(mean(data_pred$y) -  mean(preds$fit))

}
```

```r
res_krls <- sim_data %>%
  map_dbl(krls_sim)

fin <- tibble(
  res = res_krls,
  estimator = rep("krls", 1000)
)
```

```r
krls_res <- read_csv(here("data", "hw5_p3.csv"))


krls_res %>%
  ggplot(aes(x = estimator, y = res)) +
  geom_boxplot() +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed")
```



10
5
0
res
krls
estimator