MY457/MY557: Causal Inference for Experimental and Observational Studies

Class 1: Randomized Experiments

The in-class exercise this week will walk through a few general principles of randomized experiments in the potential outcomes framework. For this, we generate a dataset and then show how the estimated treatment effect differs under randomized and non-randomized treatment assignment.

First, we will load in some required packages:

```
library(dplyr)
library(ggplot2)
set.seed(82732)
```

Confounding

In causal inference, we are generally interested in examining the effect of a key explanatory variable (which we often refer to as the "treatment") on an outcome of interest. We are in general attempting to identify the effect that this key explanatory variable has on the outcome (the so-called "treatment effect"), to the exclusion of any plausible competing explanations. If we cannot reasonably exclude all possible competing explanations for variation in an outcome that we want to attribute to the key explanatory variable, then we have a threat to inference known as a *confounding explanation*.

Generate a dataset

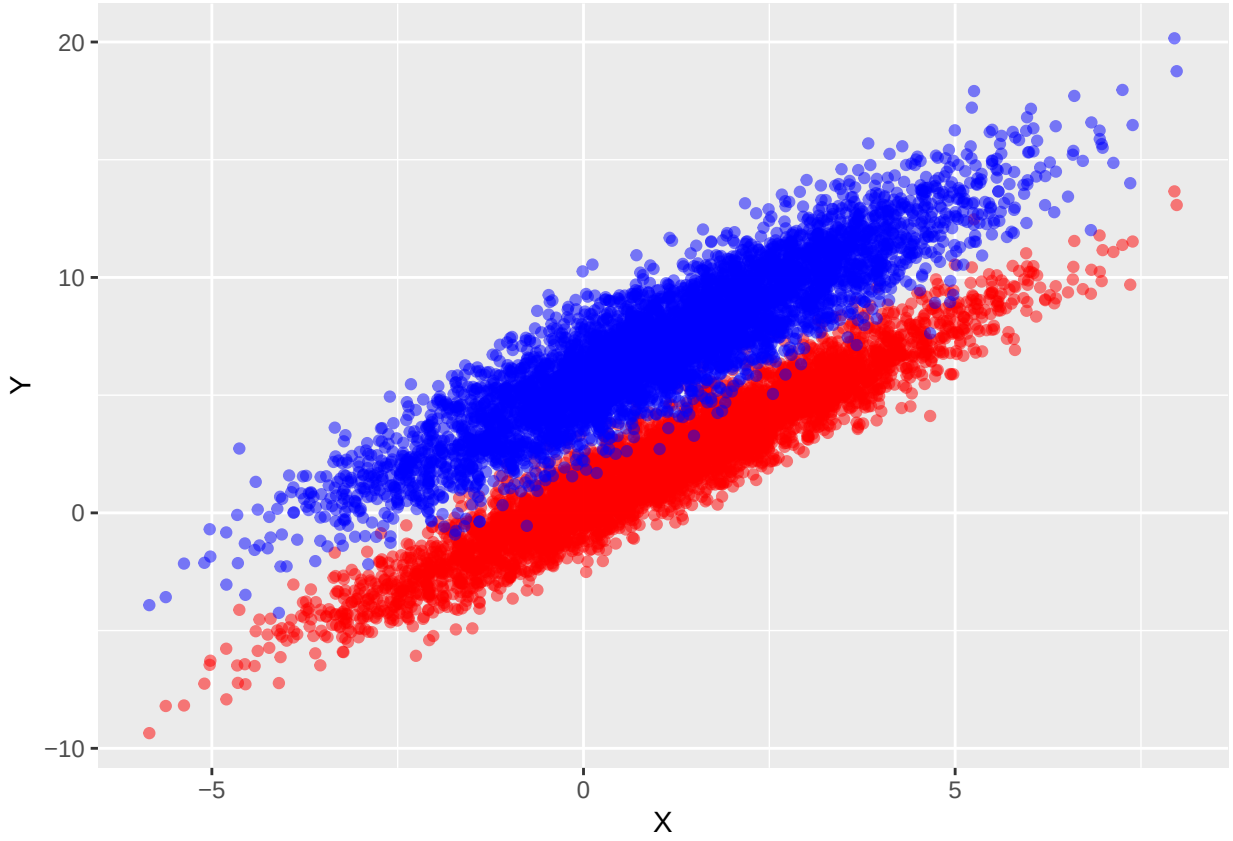
Now we will simulate some data to illustrate some general principles about experimental research within the potential outcomes framework. Then we generate a population of size $N = 5000$, with potential outcomes Y_0 and Y_1 constructed as a linear function of the measurement X_1, which is drawn from a normal distribution with a mean of 1 and standard deviation of 2. Y_0 is a linear function of X_1, parameters, and random error, while Y_1 is an additive function of Y_0, a constant, and random error.

```
# DEFINE PARAMETERS
N <- 5000
treatment_effect <- 5
intercept <- 0.5
beta1 <- 1.5

# DEFINE POTENTIAL OUTCOMES
X <- rnorm(N, mean = 1, sd = 2)
Y0 = intercept + beta1*X + rnorm(N, mean = 0, sd = 1)
Y1 = Y0 + treatment_effect + rnorm(N, mean = 0, sd = 1)

# COMBINE TO DATAFRAME
df <- data.frame(Y0 = Y0, Y1 = Y1, X = X, stringsAsFactors = F)
```

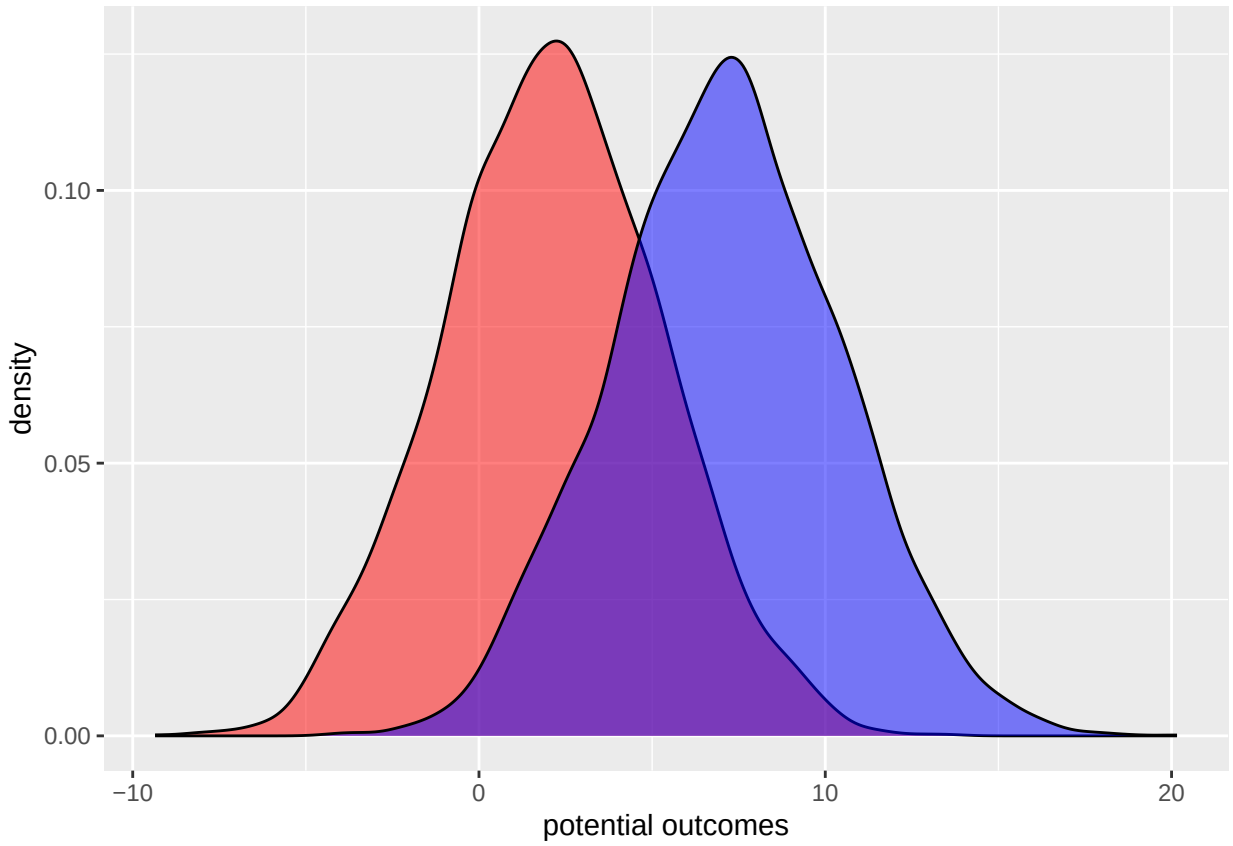
First we can visualize the potential outcomes.

```
# PLOT OF DATA GENERATING PROCESS FOR TREATMENT AND CONTROL GROUP
ggplot() +
```

```
  geom_jitter(data = df, aes(x = X, y = Y0), color = 'red', alpha = 0.5) +
  geom_jitter(data = df, aes(x = X, y = Y1), color = 'blue', alpha = 0.5) +
  xlab('X') + ylab('Y')
```




```
# DENSITY PLOT OF TREATMENT AND CONTROL GROUP
ggplot() +
  geom_density(data = df, aes(x = Y0), fill = 'red', alpha = 0.5) +
  geom_density(data = df, aes(x = Y1), fill = 'blue', alpha = 0.5) +
  xlab('potential outcomes')
```



Study the average treatment effect (ATE)

a) with potential outcomes

If we knew both potential outcomes for each case, then we would be able to calculate the average treatment effect. To calculate the ATE for potential outcomes, there are two ways to do this: (1) take the simple difference in the means of Y_1 and Y_0; or (2) take the average of the individual treatment effects

```
# 1. simple difference in means
mean(df$Y1)-mean(df$Y0)
```

```
## [1] 5.004377
```

```
# 2. average of individual treatment effects
df <- df %>% mutate(teffect = Y1 - Y0)
mean(df$teffect)
```

```
## [1] 5.004377
```

b) with real outcomes

However, recall that our ability to see both potential outcomes for a case only exists in the world of simulations. In real-world data, we will only see an outcome called Y, which is either Y_0 or $Y1$ depending on the value of the treatment indicator D. There are two possible scenarios: (1) D is randomly assigned, vs. (2) D is not randomly assigned. Let's start with the first case that D is randomly assigned to each unit.

```
# CREATE TREATMENT INDICATOR AND ACTUAL OUTCOME
df <- df %>%
  mutate(D_random = rbinom(nrow(.), 1, 0.5)) %>%
  mutate(Y = case_when(D_random == 1 ~ Y1, D_random == 0 ~ Y0))
```

Now, we can calculate the average treatment effect (ATE). There are two ways to do this: (1) by taking the simple difference in actual outcomes of treatment and control group; or (2) by regressing the real outcomes on the treatment assignment indicator.

```
# 1. simple difference in means
y_D1_mean <- df %>% subset(., D_random == 1) %>% pull(Y) %>% mean(., na.rm = T)
y_D0_mean <- df %>% subset(., D_random == 0) %>% pull(Y) %>% mean(., na.rm = T)
y_D1_mean - y_D0_mean
```

```
## [1] 5.086365
```

```
# 2. regression
lm(Y ~ D_random, data = df) %>% summary()
```

```
##
## Call:
## lm(formula = Y ~ D_random, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0899  -2.1938  -0.0162   2.1841  12.9863
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.08506    0.06417   32.49   <2e-16 ***
## D_random     5.08637    0.09084   55.99   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.212 on 4998 degrees of freedom
## Multiple R-squared:  0.3855, Adjusted R-squared:  0.3854
## F-statistic:  3135 on 1 and 4998 DF,  p-value: < 2.2e-16
```

However, what happens if D is not randomly assigned? For instance, what happens if higher values of X are more likely to receive treatment that lower values of X?

```
# CREATE TREATMENT INDICATOR AND ACTUAL OUTCOME
threshold_x <- median(X)
df <- df %>%
  mutate(D_nonrandom = case_when(X >= threshold_x ~ rbinom(n = nrow(.), size = 1, prob = .75),
                                 X < threshold_x ~ rbinom(n = nrow(.), size = 1, prob = .25))) %>%
  mutate(Y_nonrandom = case_when(D_nonrandom == 1 ~ Y1, D_nonrandom == 0 ~ Y0))
```

Again, we can calculate the "average treatment effect (ATE)" by taking the simple difference in actual outcomes of treatment and control group.

```
# ESTIMATE TREATMENT EFFECT
y_D1_mean <- df %>% subset(., D_nonrandom == 1) %>% pull(Y_nonrandom) %>% mean(., na.rm = T)
y_D0_mean <- df %>% subset(., D_nonrandom == 0) %>% pull(Y_nonrandom) %>% mean(., na.rm = T)
y_D1_mean - y_D0_mean
```

```
## [1] 7.43523
```

Quite a difference, huh? In other words, when treatment assignment is not random, then we have a classic confounding problem. Put differently, if Y and D are both affected by another factor (e.g. X), then we cannot simply take the difference in mean outcomes between treatment and control group because we have not corrected for the confounder.

The size of the confound depends on how strongly D is correlated with the confounder X. Let's see two

examples where the correlation between D and X is high and low. We start with the low one by setting the probability of receiving treatment to a roughly similar level.

```
# Propability: 55% vs. 45%

# CREATE TREATMENT INDICATOR AND ACTUAL OUTCOME
threshold_x <- median(X)
df <- df %>%
  mutate(D_nonrandom = case_when(X >= threshold_x ~ rbinom(n = nrow(.), size = 1, prob = .55),
                                 X < threshold_x ~ rbinom(n = nrow(.), size = 1, prob = .45))) %>%
  mutate(Y_nonrandom = case_when(D_nonrandom == 1 ~ Y1, D_nonrandom == 0 ~ Y0))

# ESTIMATE TREATMENT EFFECT
y_D1_mean <- df %>% subset(., D_nonrandom == 1) %>% pull(Y_nonrandom) %>% mean(., na.rm = T)
y_D0_mean <- df %>% subset(., D_nonrandom == 0) %>% pull(Y_nonrandom) %>% mean(., na.rm = T)
y_D1_mean - y_D0_mean
```

```
## [1] 5.406787
```

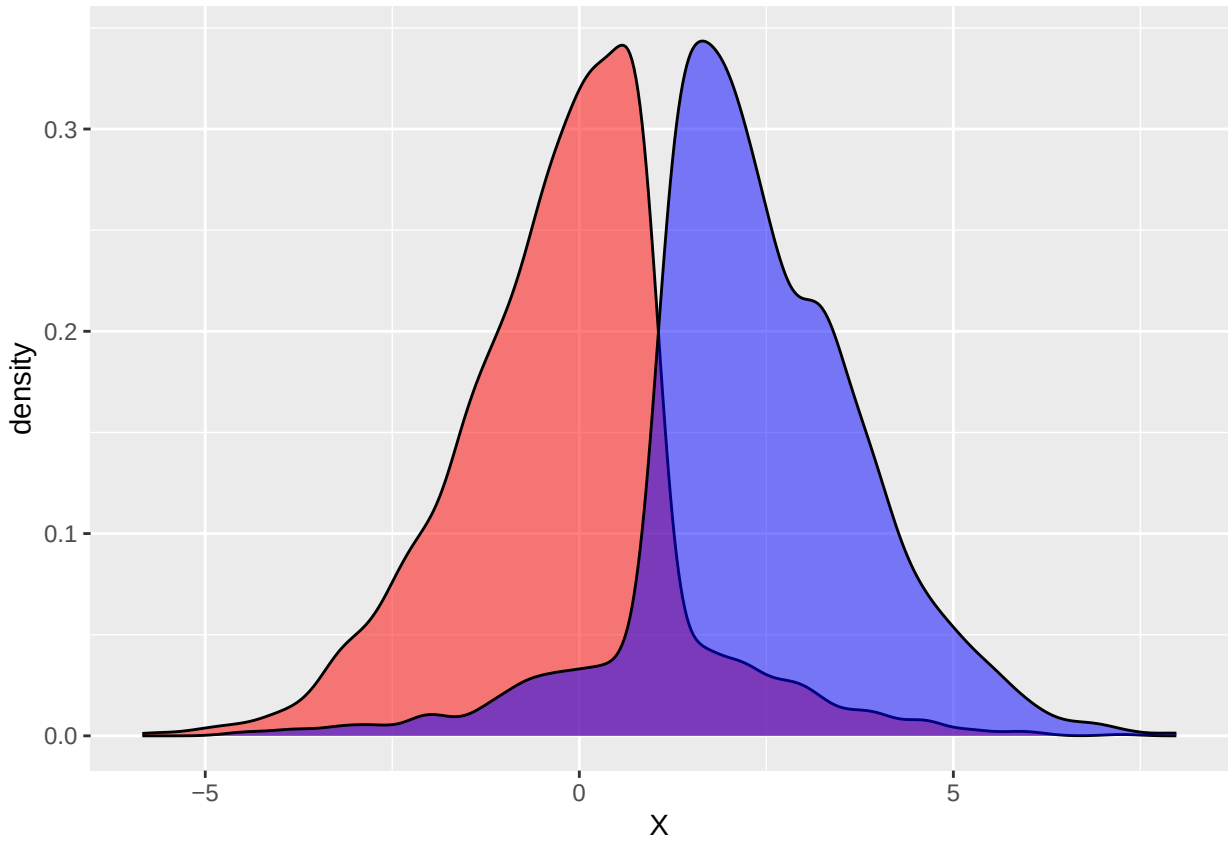
As we can see, the estimated treatment effect is not too different from the true ATE. However, it is still biased. But what happens if the probability of receiving treatment depends very strongly on the values of X. Let's see an example:

```
# Propability: 90% vs. 10%

# CREATE TREATMENT INDICATOR AND ACTUAL OUTCOME
threshold_x <- median(X)
df <- df %>%
  mutate(D_nonrandom = case_when(X >= threshold_x ~ rbinom(n = nrow(.), size = 1, prob = .90),
                                 X < threshold_x ~ rbinom(n = nrow(.), size = 1, prob = .10))) %>%
  mutate(Y_nonrandom = case_when(D_nonrandom == 1 ~ Y1, D_nonrandom == 0 ~ Y0))

# ESTIMATE TREATMENT EFFECT
y_D1_mean <- df %>% subset(., D_nonrandom == 1) %>% pull(Y_nonrandom) %>% mean(., na.rm = T)
y_D0_mean <- df %>% subset(., D_nonrandom == 0) %>% pull(Y_nonrandom) %>% mean(., na.rm = T)
y_D1_mean - y_D0_mean
```

```
## [1] 8.870918
```

The reason for the big difference between the biased estimate and the true ATE is due to fact that the treatment group consists primarily of units that have high X. We can see the big difference between the treatment and control group when we plot the density of X for both groups.

```
# DENSITY PLOT OF X FOR TREATMENT AND CONTROL GROUP
ggplot() +
  geom_density(data = df %>% subset(., D_nonrandom == 0), aes(x = X), fill = 'red', alpha = 0.5) +
  geom_density(data = df %>% subset(., D_nonrandom == 1), aes(x = X), fill = 'blue', alpha = 0.5) +
  xlab('X')
```



Can we not just control for it?

Okay, let's do this: we regress the actual outcome Y on the non-randomized treatment assignment indicator D while controlling for X.

```
lm(Y_nonrandom ~ D_nonrandom + X, data = df) %>% summary()
```

```
##
## Call:
## lm(formula = Y_nonrandom ~ D_nonrandom + X, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9002 -0.7932 -0.0058  0.7850  4.4976
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.54013    0.02424   22.28   <2e-16 ***
## D_nonrandom  4.95948    0.04506  110.06   <2e-16 ***
## X            1.50727    0.01134  132.96   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.207 on 4997 degrees of freedom
## Multiple R-squared:  0.9446, Adjusted R-squared:  0.9446
## F-statistic: 4.261e+04 on 2 and 4997 DF,  p-value: < 2.2e-16
```

As we can see, when we control for X, we get quite close to the true ATE. Also, the coefficient for X reflects

the parameter in the data generating process quite well. However, there is still some bad news: we (almost) never know the data generating process in real-world settings, and thus, we do not know whether (and how) X affects D and Y. And even if we knew about X, we still do not know all the other (unobservable) factors that can also affect D and Y. Therefore, we need other approaches, which we will learn throughout this course.

Assessing whether randomization "worked"

a) sample size

We have seen that with a sample of 5000 observations, the estimated ATE with randomized treatment assignment is quite close to the ATE that we used when we simulated the data (in the data generating process). However, what happens if we were to decrease the sample size? Does the estimate deviate more strongly from the true ATE?

```
# FUNCTION TO SIMULATE DATASET WITH VARYING SIZE
simulateDataAndATE <- function(N){
  treatment_effect <- 5
  intercept <- 0.5
  beta1 <- 1.5
  X <- rnorm(N, mean = 1, sd = 2)
  Y0 = intercept + beta1*X + rnorm(N, mean = 0, sd = 1)
  Y1 = Y0 + treatment_effect + rnorm(N, mean = 0, sd = 1)
  df <- data.frame(Y0 = Y0, Y1 = Y1, X = X, stringsAsFactors = F) %>%
    mutate(D_random = rbinom(nrow(.), 1, 0.5)) %>%
    mutate(Y = case_when(D_random == 1 ~ Y1, D_random == 0 ~ Y0))

  y_D1_mean <- df %>% subset(., D_random == 1) %>% pull(Y) %>% mean(., na.rm = T)
  y_D0_mean <- df %>% subset(., D_random == 0) %>% pull(Y) %>% mean(., na.rm = T)
  teffect <- y_D1_mean - y_D0_mean

  paste('number of observations: ', N, " | true ATE: 5 | estimated ATE: ", teffect, sep = '') %>% print
}

# RUN LOOP
for(n in c(10, 20, 30, 50, 100, 150, 1000, 2000, 5000, 10000, 20000, 50000)){
  simulateDataAndATE(N = n)
}
```

```
## [1] "number of observations: 10 | true ATE: 5 | estimated ATE: 6.83195603883853"
## [1] "number of observations: 20 | true ATE: 5 | estimated ATE: 7.20319851553929"
## [1] "number of observations: 30 | true ATE: 5 | estimated ATE: 5.50287601634448"
## [1] "number of observations: 50 | true ATE: 5 | estimated ATE: 3.9907211874983"
## [1] "number of observations: 100 | true ATE: 5 | estimated ATE: 4.28209388212644"
## [1] "number of observations: 150 | true ATE: 5 | estimated ATE: 4.89508985709694"
## [1] "number of observations: 1000 | true ATE: 5 | estimated ATE: 4.85209178845315"
## [1] "number of observations: 2000 | true ATE: 5 | estimated ATE: 4.86445772418768"
## [1] "number of observations: 5000 | true ATE: 5 | estimated ATE: 4.84647166203712"
## [1] "number of observations: 10000 | true ATE: 5 | estimated ATE: 4.97492505846575"
## [1] "number of observations: 20000 | true ATE: 5 | estimated ATE: 4.97349252944939"
## [1] "number of observations: 50000 | true ATE: 5 | estimated ATE: 4.99196091944938"
```

While the averages of the estimated treatment effects for each sample size is somewhat close to 5 (the true ATE), we see that the estimates for the smaller sample sizes can differ more from the true ATE than the estimates that are based on the larger sample sizes. In other words, although the estimates of the treatment effects based on small sample sizes are still unbiased, a larger sample size lowers sampling uncertainty and

increase precision in the estimates.

b) balance tests

In a real-world data analysis setting, we will never know in practice how well random assignment has removed confounding by balancing the treatment groups in all possible confounding variables. While the theory tells us that it does remove confounding in expectation, in actual finite datasets some imbalance will remain because of randomization variability. In general we can say though that the larger the dataset, the better.

In addition, it is always good to examine the balance between experimental groups with respect to some measured covariates. If randomization did what it was intended to do, we should see broadly similar distributions of all factors that we can measure between those units that received treatment and those units that did not.

```
# DENSITY PLOT OF X FOR TREATMENT AND CONTROL GROUP
ggplot() +
  geom_density(data = df %>% subset(., D_random == 0), aes(x = X), fill = 'red', alpha = 0.5) +
  geom_density(data = df %>% subset(., D_random == 1), aes(x = X), fill = 'blue', alpha = 0.5) +
  xlab('X')
```