ECON 773: Assignment 2

The BLUE Team, Jeneta Ljutic (400138620), Tadhg Taylor-McGreal (400330297), Stella Till (400364649)

2026-01-19

Table of contents

Preface

Goal

The goals of this assignment are to:

- explore the properties of the OLS estimator using Monte Carlo experiments
- analyze data from an RCT using `lm_robust` and `lm_lin`

Instructions

See assignment 1.

Monte Carlo experiments

Monte Carlo 1: coin flips

This question will be demonstrated in class, as an introduction to Monte Carlo experiments.

Flip a coin once (`1` for heads, `0` for tails), and store in `X_1`:

```
(X_1 <- sample(0:1, size = 1, replace = TRUE))
```

```
[1] 0
```

The variable `n` will refer to sample size. Roll a dice `n` times:

```
n <- 10
(X_n <- sample(0:1, size = n, replace = TRUE))
```

```
 [1] 1 0 1 1 1 1 1 1 1 1
```

We will use a large set of packages for this assignment, loaded in the next chunk. Do you recognize them?

```
install.packages("gtsummary")
install.packages("gt")
install.packages("gtsummary")
install.packages("estimatr")
install.packages("AER")
```

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0
──
✔ dplyr     1.1.4     ✔ readr     2.1.6
✔ forcats   1.0.1     ✔ stringr   1.6.0
✔ ggplot2   4.0.1     ✔ tibble    3.3.0
✔ lubridate 1.9.4     ✔ tidyr     1.3.1
✔ purrr     1.2.0
── Conflicts ────────────────────────────────────────── tidyverse_conflicts()
──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```r
library(gt)
library(gtsummary)
library(estimatr)
library(broom)
```

To compute the mean up to and including the ith throw, we can use the `cummean` function from `dplyr`:

```r
Xbar_in <- cummean(X_n)
cbind(X_n, Xbar_in)
```

```
      X_n   Xbar_in
 [1,]   1 1.0000000
 [2,]   0 0.5000000
 [3,]   1 0.6666667
 [4,]   1 0.7500000
 [5,]   1 0.8000000
 [6,]   1 0.8333333
 [7,]   1 0.8571429
 [8,]   1 0.8750000
 [9,]   1 0.8888889
[10,]   1 0.9000000
```

If you only want the mean across all `n` flips, can do:

```r
(Xbar_n <- mean(sample(0:1, size = n, replace = TRUE)))
```

```
[1] 0.6
```

We can use the following function to generate a tibble that keeps track across flips:

```r
gen_coin_cm <- function(n, experiment) {
    X_n <- sample(0:1, size = n, replace = TRUE)
    Xbar_in <- cummean(X_n)
    return(tibble(i = 1:n, X_n = X_n, Xbar_in = Xbar_in, experiment =
experiment))
}
```

Once you define a function, you can use it as follows:

```r
gen_coin_cm(20, 3)
```

```
# A tibble: 20 × 4
       i   X_n Xbar_in experiment
   <int> <int>   <dbl>      <dbl>
 1     1     1   1              3
 2     2     1   1              3
 3     3     1   1              3
 4     4     1   1              3
 5     5     0   0.8            3
 6     6     1   0.833          3
 7     7     0   0.714          3
 8     8     0   0.625          3
 9     9     0   0.556          3
10    10     0   0.5            3
11    11     1   0.545          3
12    12     1   0.583          3
13    13     1   0.615          3
14    14     1   0.643          3
15    15     0   0.6            3
16    16     1   0.625          3
17    17     0   0.588          3
18    18     0   0.556          3
19    19     1   0.579          3
20    20     1   0.6            3
```

First, generate a tibble `coin_df` with `n = 10` and `experiment = 1`. Second, make a plot of `coin_df` with `i` on the horizontal and `Xbar_in` on the vertical. Third, set `n = 1000`, generate `coin1_df` as in part 1 of this question, and redo the plot in part 2. Fourth, generate additional data sets `coin2_df` and `coin3_df` as in part 3 of this question. Join them all together using `bind_rows`. Then, make a line plot with a different colour for each experiment, using `colour = factor(experiment)` in `geom_line`. Fifth, make one more plot with one experiment and `n = 100000`.

Interpret your results. Is the phenomenon you observe related to the LLN or the CLT?

Answer

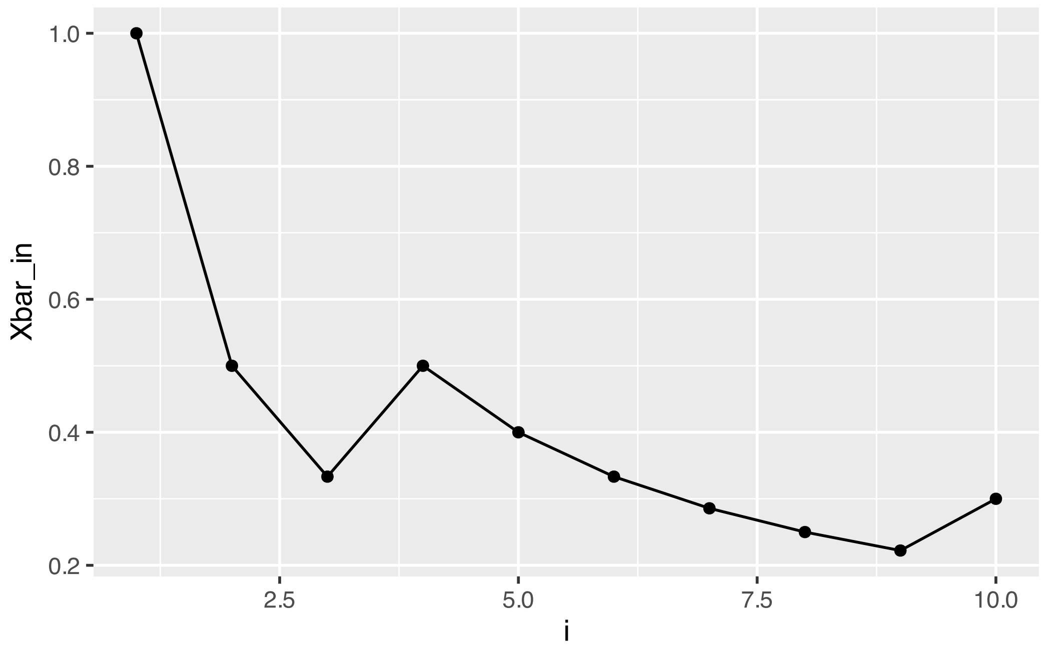
1. We are going to use the 'gen_coin_cm' function that is given in the question, to simulate 10 coin flips from experiment 1.

```r
coin_df <- gen_coin_cm(10,1)
```

This resulted in a data frame containing 10 coin flips including the trial number, the result of the i-th coin flip, the cumulative mean up to trial i, and a label identifying the experiment.

2. In order to see whether this worked, we now make the plot requested in the second part of this question.

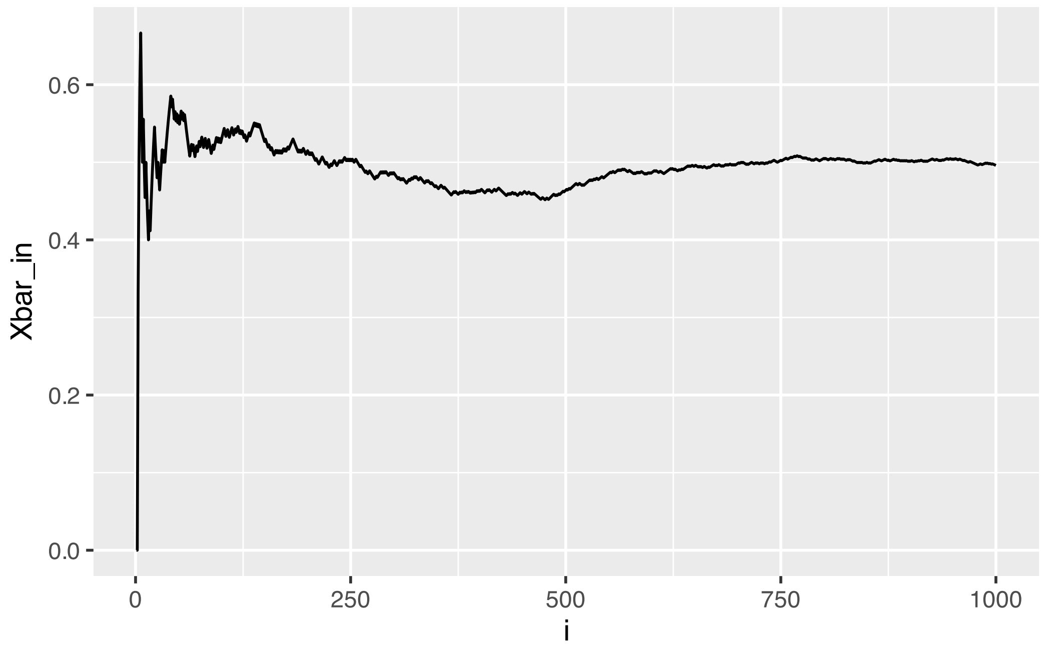
```r
coin_df |> ggplot() +
    geom_point(aes(x=i, y=Xbar_in)) +
    geom_line(aes(x=i, y=Xbar_in))
```



This plot shows the cumulative mean of the coin flips as a function of the trial number, illustrating how the sample mean evolves over time as additional trials are added.

3. To compelte the third part of this question, we are going to copy, paste and modifiy the code from above.

```r
coin1_df <- gen_coin_cm(1000,1)
coin1_df |> ggplot() +
    geom_line(aes(x=i, y=Xbar_in))
```



We simulated 1,000 coin flips and plotted how the running average changes over time. With a larger number of trials, we observe some convergence of the sample mean toward 0.5.

4. Next, we repeat this experiment twice more.

```
coin2_df <- gen_coin_cm(1000,2)
coin3_df <- gen_coin_cm(1000,3)
```

Using the same function and number of trials, we have created two independent simulated experiments.

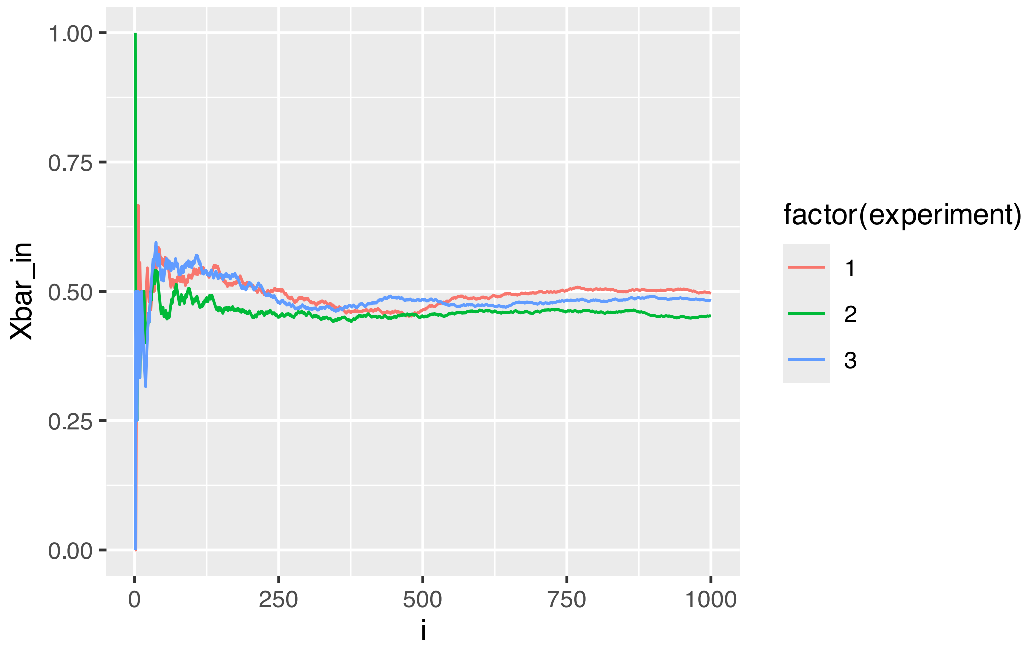
Put all the experimental data on a big pile.

```
all_experiments <- bind_rows(
    coin1_df, coin2_df, coin3_df
)
```

'all_experiments' now contains each of the three simulated experiments.

Now we are ready to make the final plot of this question.

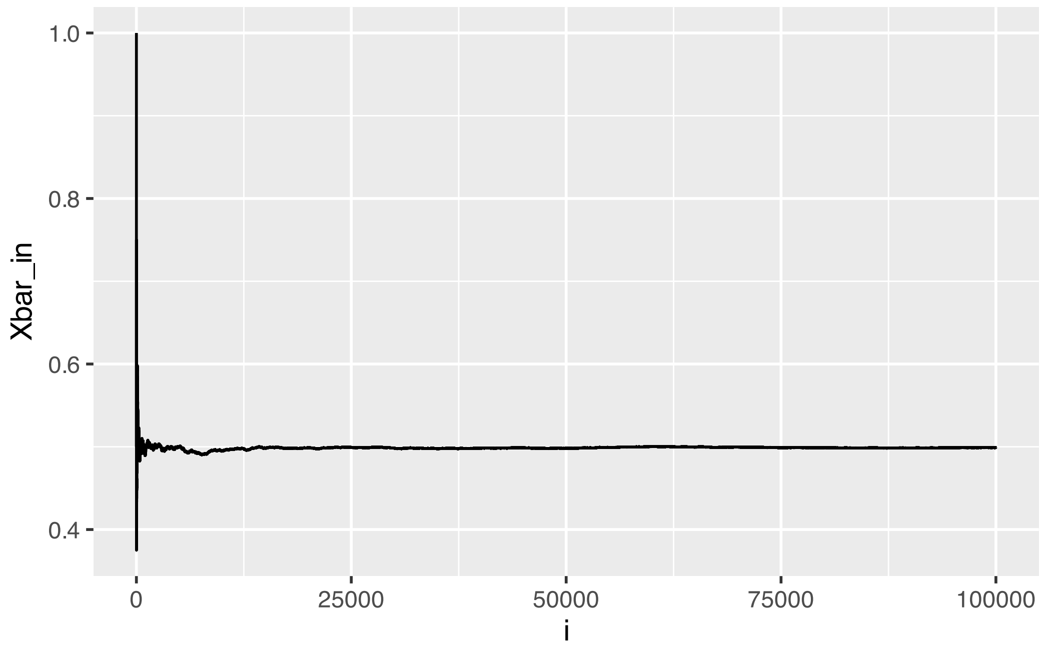
```
all_experiments |> ggplot() +
    geom_line(aes(x=i, y=Xbar_in, colour=factor(experiment)))
```



Each colour represents one of the 3 experiments. 1000 trials is not enough for full convergence to 0.5.

5. Using the same function, we will create one more plot with 100000 trials.

```
big_coin_df <- gen_coin_cm(100000,1)
big_coin_df |> ggplot() +
geom_line(aes(x=i, y=Xbar_in))
```



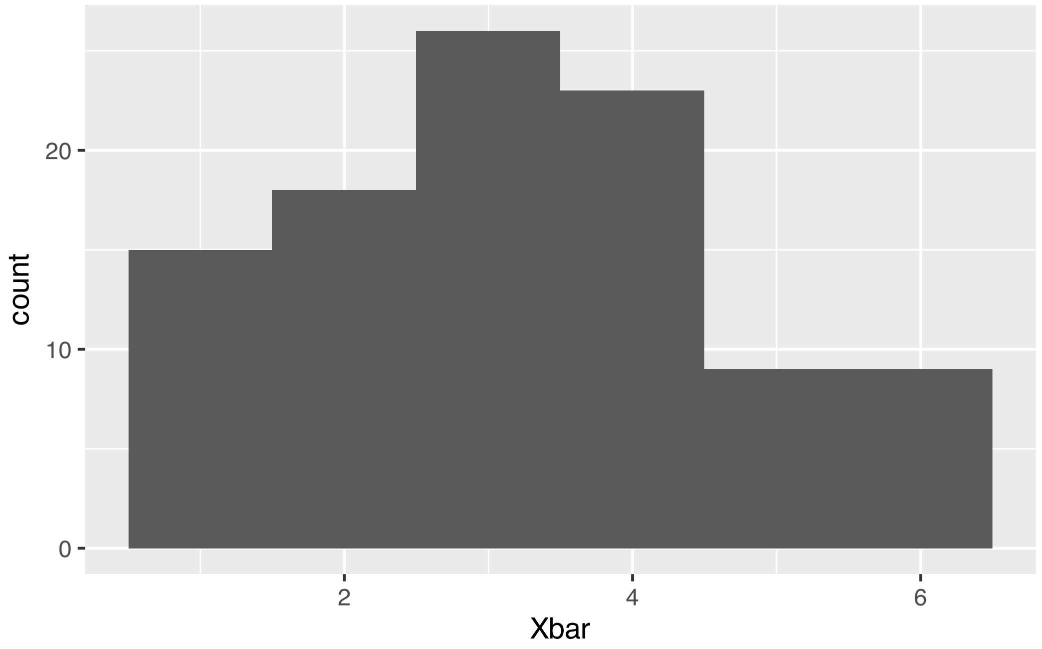
This shows the Law of Large Numbers (LLN). The LLN states that the sample average X, converts to the true mean E[X] = 0.5 as n approaches infinity. We see this as the line graph converges towards 0.5. CLT describes that the distribution of the sample average of X centers around 0.5 for large averages. This is not indicated on the line plot.

Monte Carlo 2: dice throws

In the previous exercise, we fixed `S` and let `n` grow large. Let us now slowly grow `n` and see what happens, letting `S` be very large. We start with `n = 1`, corresponding to one roll of the die. (For this second exercise, we will use a die instead of a coin.)

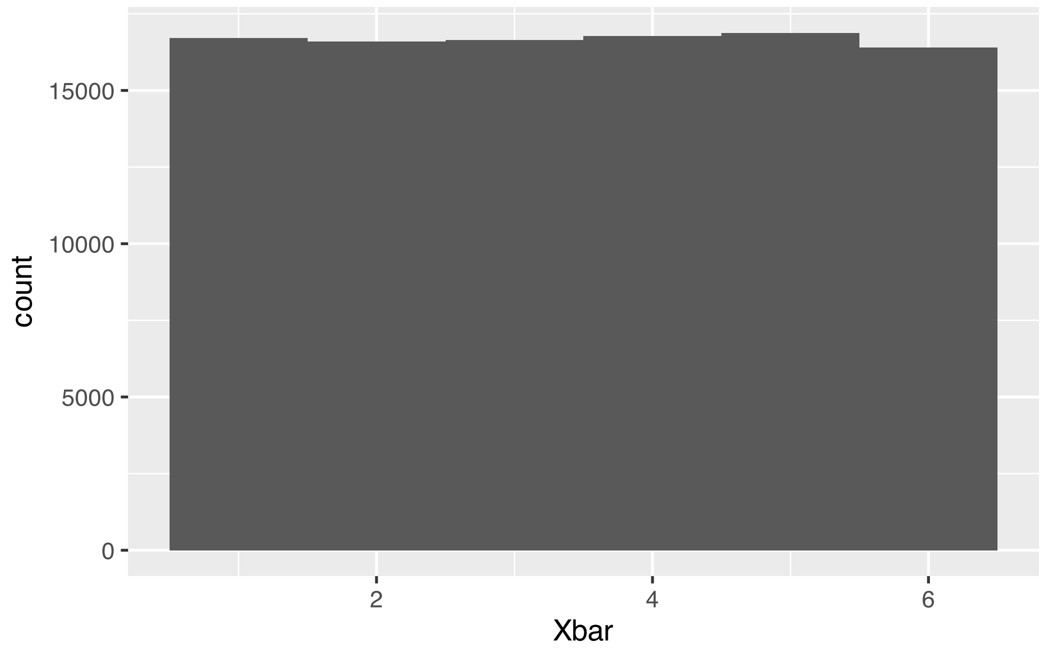
One roll:

```r
S <- 100
x <- sample(1:6, S, replace = TRUE)
die_df <- tibble(s = 1:S, Xbar = x)
die_df |> ggplot(aes(x = Xbar)) +
    geom_histogram(binwidth = 1)
```



Let us redo this with larger S.

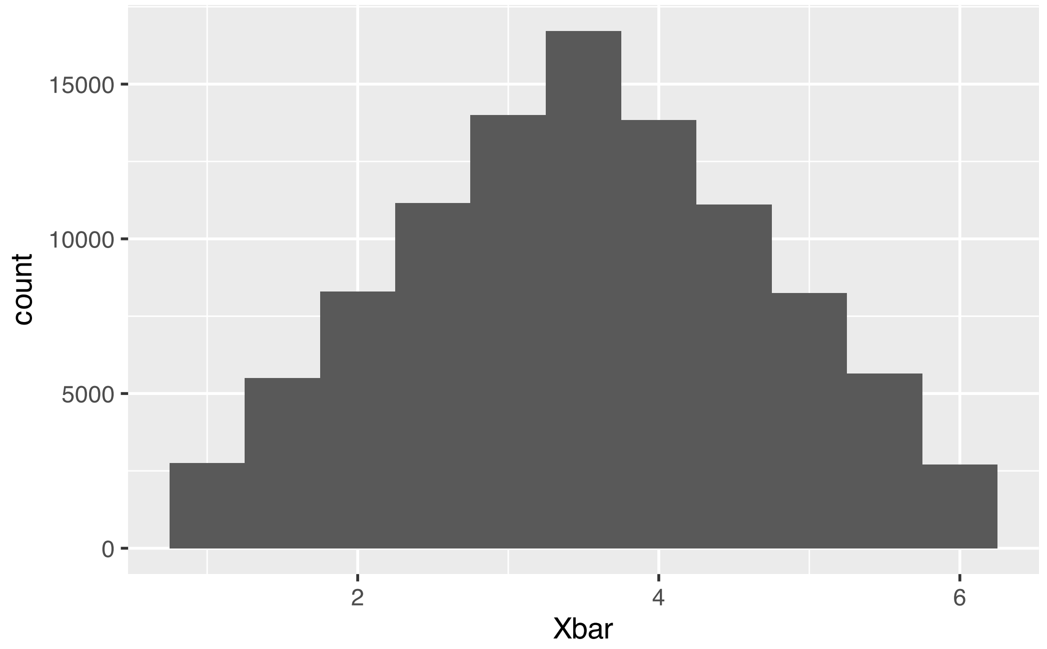
```
S <- 100000
x <- sample(1:6, S, replace = TRUE)
die_df <- tibble(s = 1:S, Xbar = x)
die_df |> ggplot(aes(x = Xbar)) +
    geom_histogram(binwidth = 1)
```



A large `S` seems to be closer to the population probabilities of $P(X = x) = 1/6$. Because our goal is to explore the theoretical properties of random variables, we will stick with large `S`.

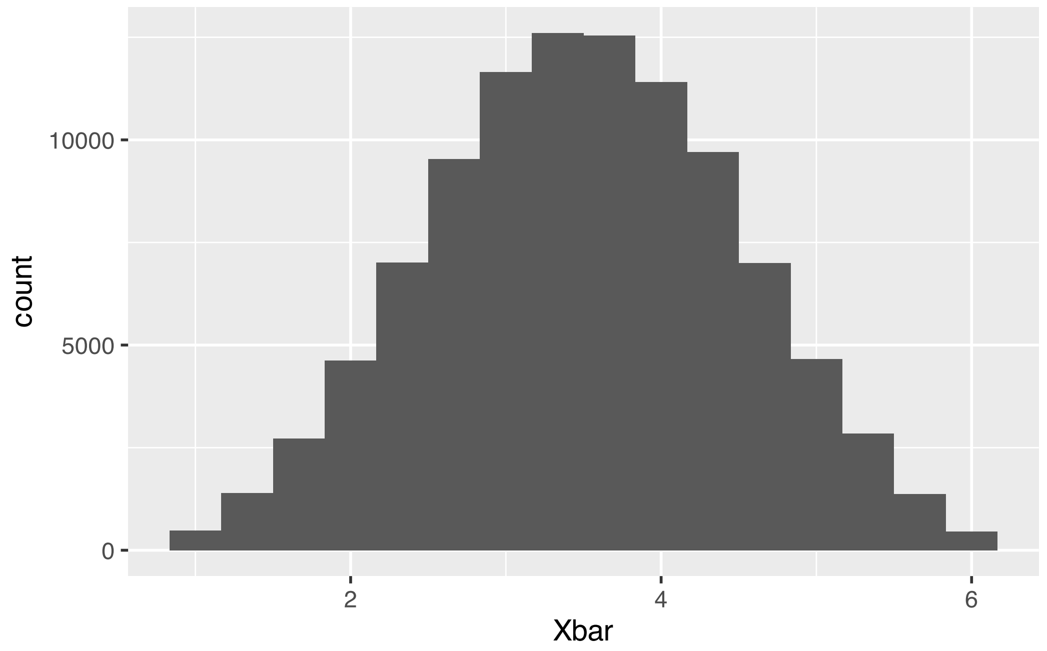
Roll twice and average.

```r
n <- 2
x <- 1 / n * (sample(1:6, S, replace = TRUE) + sample(1:6, S, replace = TRUE))
die_df <- tibble(s = 1:S, Xbar = x)
die_df |> ggplot(aes(x = Xbar)) +
    geom_histogram(binwidth = 1 / n)
```



Roll three times and average.

```
n <- 3
x <- 1 / n * sample(1:6, S, replace = TRUE)
for (k in 2:n) {
    x <- x + 1 / n * sample(1:6, S, replace = TRUE)
}
die_df <- tibble(s = 1:S, Xbar = x)
die_df |> ggplot(aes(x = Xbar)) +
    geom_histogram(binwidth = 1 / n)
```



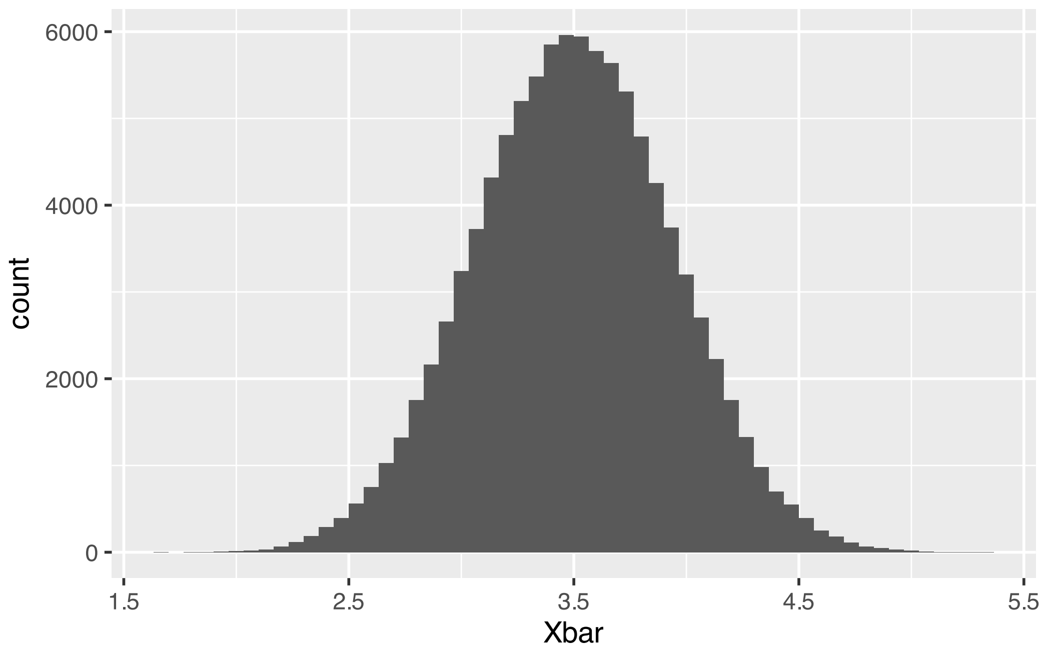
Experiment with a few more values of n (let n = 50 be your largest one.)

First: what happens to the variance of the mean as n increases? Second: what happens to the shape of the distribution as n increases? Relate your answer to the Central Limit Theorem (CLT).

Answer

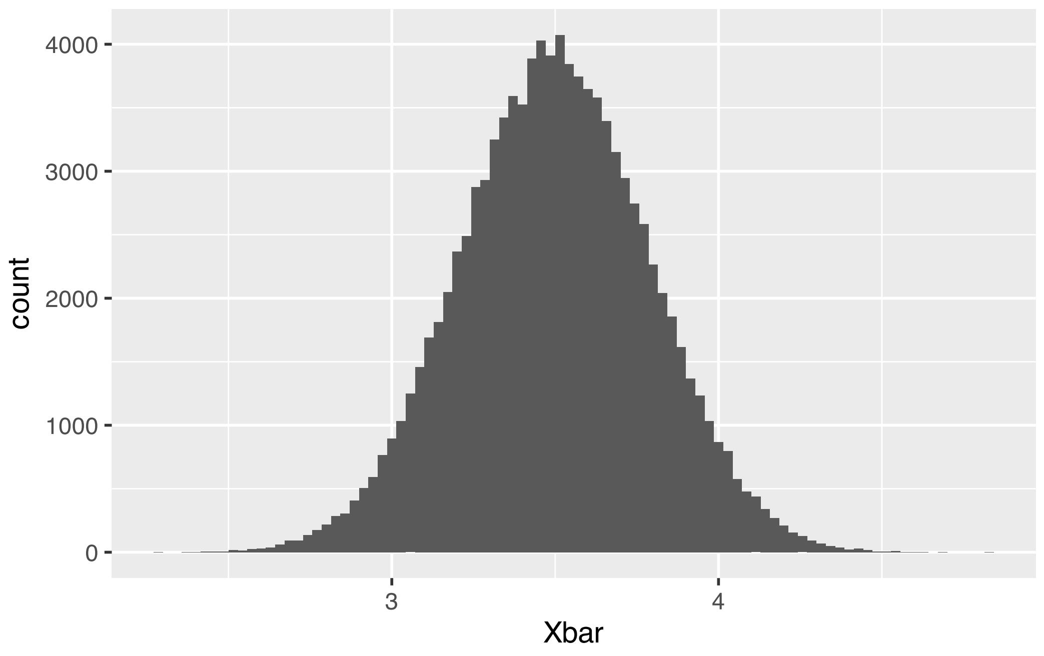
We will experiment with a few more values of n. We'll try values for n=15, n=35 and n=50, using the same code as above. Starting with n=15:

```r
n <- 15
x <- 1 / n * sample(1:6, S, replace = TRUE)
for (k in 2:n) {
x <- x + 1 / n * sample(1:6, S, replace = TRUE)
}
die_df <- tibble(s = 1:S, Xbar = x)
die_df |> ggplot(aes(x = Xbar)) +
geom_histogram(binwidth = 1 / n)
```



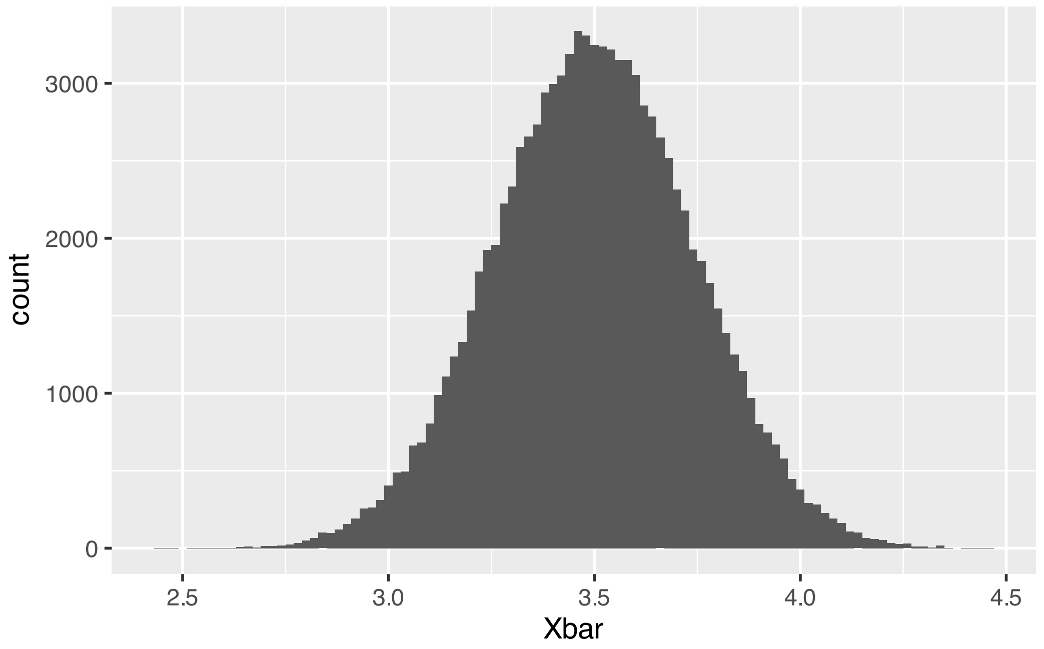
Next, n=35:

```
n <- 35
x <- 1 / n * sample(1:6, S, replace = TRUE)
for (k in 2:n) {
    x <- x + 1 / n * sample(1:6, S, replace = TRUE)
}
die_df <- tibble(s = 1:S, Xbar = x)
die_df |> ggplot(aes(x = Xbar)) +
    geom_histogram(binwidth = 1 / n)
```



Lastly, n=50:

```
n <- 50
x <- 1 / n * sample(1:6, S, replace = TRUE)
for (k in 2:n) {
    x <- x + 1 / n * sample(1:6, S, replace = TRUE)
}
die_df <- tibble(s = 1:S, Xbar = x)
die_df |> ggplot(aes(x = Xbar)) +
    geom_histogram(binwidth = 1 / n)
```



1. The variance of the mean shrinks when the number of rolls (n), increases. We can interpret the change in the variance by looking at each plot. First, we see that the distribution of the plot with the smallest n is almost flat, this visually shows the sample mean of the die roll. Because the spread is wide and tightly clustered, the variance is large. The distribution over [1,6] is almost uniform. Here we do not see any effect from CLT.
2. Next, when n=6, we see that extreme values at the ends of the distribution start to flatten. This trend continues as n increases. This shows the CLT. In a large sample, we see CLT as the shape becomes approximately normal showing a bell shape around the average, 3.5.

The Law of Large Numbers explains the center moving toward the average value 3.5 showing that the sample mean converges to the theoretical expected value.

Monte Carlo 3: Inference in RCTs

This question will be solved in class.

We want to understand the properties of different estimators in an RCT. We consider a setup where the treatment effect varies with a covariate X, and treatment assignment is unbalanced.

- $X_i \in \{0, 1\}$ with $P(X_i = 1) = 0.5$
- $D_i \in \{0, 1\}$ with $P(D_i = 1 \mid X_i) = 0.2$
- Outcomes are generated as:

  $$Y_i = 0 + 1 \cdot D_i + 5 \cdot X_i + 5 \cdot (D_i \times X_i) + \epsilon_i$$

  where $\epsilon_i \sim N\left(0, (1 + 2X_i)^2\right)$.

Calculate the ATE and the CATEs for $X = 0$ and $X = 1$ based on these parameters.

We want to see if the reported standard errors from simple linear regressions are trustworthy. We compare two common approaches:

1. DIM: `lm(Y ~ D)`.
2. Additive: `lm(Y ~ D + X)`.

We run a simulation to check:

1. Bias: Is the average estimate close to the true ATE?
2. Efficiency: What is the standard deviation of the estimates?
3. Coverage Probability: How often does the 95% Confidence Interval contain the true ATE?

Consider the following function that performs one simulation replication and returns a tibble with the results for both estimators:

```r
dgp_het <- function(n) {
    D <- rbinom(n, 1, 0.2) # Unbalanced assignment
    X <- rbinom(n, 1, 0.5)
    e <- rnorm(n, 0, 1 + 2 * X) # Heteroskedastic errors
    Y <- 0 + 1 * D + 5 * X + 5 * D * X + e
    return(tibble(D = D, X = X, Y = Y))
}

sim_one <- function(s, n) {
    df <- dgp_het(n)
    truth <- 3.5

    # 1. DIM
    fit_dim <- lm(Y ~ D, data = df)
    res_dim <- tidy(fit_dim, conf.int = TRUE) |>
        filter(term == "D") |>
        mutate(estimator = "DIM", covered = (conf.low < truth & conf.high >
truth))
```

```r
    # 2. Additive
    fit_add <- lm(Y ~ D + X, data = df)
    res_add <- tidy(fit_add, conf.int = TRUE) |>
        filter(term == "D") |>
        mutate(
            estimator = "Additive",
            covered = (conf.low < truth & conf.high > truth)
        )

    bind_rows(res_dim, res_add) |>
        mutate(s = s)
}

sim_one(1, 200)
```

```
# A tibble: 2 × 10
  term  estimate std.error statistic  p.value conf.low conf.high estimator
  <chr>    <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl> <chr>
1 D         4.11     0.717      5.73 3.76e- 8     2.69      5.52 DIM
2 D         3.71     0.469      7.92 1.75e-13     2.79      4.63 Additive
# ℹ 2 more variables: covered <lgl>, s <dbl>
```

Your task is to:

1. Run this simulation $S = 1000$ times with $n = 200$.
2. Summarize the results: calculate the mean estimate, standard deviation of the estimate, and mean coverage probability for each estimator.
3. Interpret the findings related to bias and efficiency of the estimators: which estimator do you prefer?
4. Interpret the findings related to coverage probability: do you trust the confidence intervals from these estimators?

Answer

1. Start by running the simulation 1000 times with n = 200 using ‘sim_one’.

```r
S <- 1000
test_df <- 1:S |> map_df(sim_one, n=200)
```

We now have a dataframe ‘test_df’ with our simulation of both approaches.

2. Next we will sumamrize the results to calculate the mean estimate, standard deviation of the estimate, and mean coverage probability for each estimator.

```r
test_df |> group_by(estimator) |>
        summarize(
            mean_ATE_hat = mean(estimate),
            sd_ATE_hat = sd(estimate),
            coverage = mean(covered)
        )
```

```
# A tibble: 2 × 4
  estimator mean_ATE_hat sd_ATE_hat coverage
  <chr>            <dbl>      <dbl>    <dbl>
1 Additive          3.48      0.521    0.896
2 DIM               3.47      0.917    0.866
```

Commentary on the later part of the questions.

3. Summary of output (Additive): Our ATE is equal to 3.52, a slight upward bias of 0.02. The standard deviation of the ATE is around 0.520. The 95% CI coverage was 0.900. This means that in the 95% confidence interval it only contained the true ATE 90% of the time. Only 90% of the simulations included the ATE = 3.5. It is likely explained that the average rate of 0.900 for a nominal 95% confidence interval means the interval is systematically too narrow and understates uncertainty. This could likely be due to incorrect standard error assumptions (e.g., ignoring heteroskedasticity).

Summary of output (DIM): Our ATE is equal to 3.53. The standard deviation of the ATE is 0.921. The 95% CI coverage was 0.854. There is a slight upward bias in the DIM (3.53 - 3.5 = 0.03). This could be explained by noise from the Monte Carlo simulation.

We see that the additive estimator is more efficient. This is because adjusting for X accounts for some of the outcome variation. This is determined by comparing the standard errors in the table (0.520 < 0.921). Therefore, we would prefer the additive estimates because of their lower standard error and stronger coverage in the 95% CI.

4. Both estimators show undercoverage: the 95% confidence intervals contain the true ATE less than 95% of the time (about 90% for Additive and 85% for DIM). This means that, as reported by lm(), the confidence intervals are too narrow and therefore not fully trustworthy.

The problem is more severe for the DIM estimator, which ignores the covariate X. Because treatment is unbalanced and errors are heteroskedastic, omitting X leads to larger residual variance and standard errors that understate uncertainty.

We do not fully trust the confidence intervals from either estimator as implemented. The Additive model is preferable, but correct coverage would likely require heteroskedasticity-robust standard errors and a specification that includes the interaction term to account for treatment effect heterogeneity (as the model ignores the interaction and the variance structure). …

Monte Carlo 4: Lin's estimator

Now it is your turn.

Replicate the simulation from Monte Carlo 3 but add a third estimator: Lin's Estimator. Use `lm_lin(Y ~ D, covariates = ~ X)`. You should then be able to extract the estimator information using something like:

```r
est_lin <- lm_lin(Y ~ D, covariates = ~X, data = df)
res_lin <- tidy(est_lin) |>
    filter(term == "D") |>
    mutate(estimator = "Lin", covered = (conf.low < truth & conf.high >
truth))
```

Questions:

1. Write a function `sim_lin_ext(s, n)` that returns a tibble with results for DIM, Additive, and Lin.
2. Calculate the mean estimate, standard deviation, and coverage probability for all three.
3. Does Lin's estimator achieve the correct coverage?
4. Which estimator has the lowest variance?

Use $S = 100$ and $n = 1000$.

Answer

1. Start off by writing a function that combines reults from the DIM, Additive and the new Lin model. The 'lm_lin' function automatically centers covariates, includes treatment and covariate interactions internally, and uses robust standard errors by default. Therefore, it adjusts for covariates X in a way that gives better precision and correct standard errors.

```r
sim_lin_ext <- function(s, n) {
  df <- dgp_het(n)
  truth <- 3.5

  # 1. DIM
  fit_dim <- lm(Y ~ D, data = df)
  res_dim <- tidy(fit_dim, conf.int = TRUE) |>
    filter(term == "D") |>
    mutate(
      estimator = "DIM",
      covered = (conf.low < truth & conf.high > truth)
    )

  # 2. Additive
  fit_add <- lm(Y ~ D + X, data = df)
  res_add <- tidy(fit_add, conf.int = TRUE) |>
    filter(term == "D") |>
    mutate(
      estimator = "Additive",
      covered = (conf.low < truth & conf.high > truth)
    )

  # 3. Lin
  fit_lin <- lm_lin(Y ~ D, covariates = ~ X, data = df)
  res_lin <- tidy(fit_lin, conf.int = TRUE) |>
    filter(term == "D") |>
    mutate(
      estimator = "Lin",
      covered = (conf.low < truth & conf.high > truth)
    )

  bind_rows(res_dim, res_add, res_lin) |>
    mutate(s = s)
}
```

When called, this function will return a simulation of each model that returns a tibble with results for DIM, Additive, and Lin.

2. We will run the simulation S= 100 times with n = 1000 and calculate the mean estimate, standard deviation, and coverage probability for all three model versions.

```r
S <- 100     # number of replications
n <- 1000    # sample size per replication

set.seed(123)

sim_results <- 1:S |>
  map_df(sim_lin_ext, n = n)

summary_results <- sim_results |>
  group_by(estimator) |>
  summarise(
    mean_ATE_hat = mean(estimate),
    sd_ATE_hat = sd(estimate),
    coverage = mean(covered),
    .groups = "drop"
  )

summary_resultsS <- 100     # number of Monte Carlo replications
n <- 1000    # sample size per replication

set.seed(123)

sim_results <- 1:S |>
  map_df(sim_lin_ext, n = n)
```

```r
summary_results <- sim_results |>
  group_by(estimator) |>
  summarise(
    mean_ATE_hat = mean(estimate),
    sd_ATE_hat = sd(estimate),
    coverage = mean(covered),
    .groups = "drop"
  )

summary_results
```

```
# A tibble: 3 × 4
  estimator mean_ATE_hat sd_ATE_hat coverage
  <chr>            <dbl>      <dbl>    <dbl>
1 Additive          3.51      0.214     0.95
2 DIM               3.54      0.376     0.92
3 Lin               3.50      0.188     0.92
```

'summary_results' returns a table with the mean estimate, standard deviation, and coverage probability for all three models.

3. Lin's coverage is 0.92, which is still slightly below the nominal 0.95, technically it does not achieve correct coverage. This is however likely due to the unbalanced assignment (20% treated) and the heteroskedasticity: the combination of unbalanced groups and high noise means that n=1000 is actually still a "small" sample for these robust calculations. In larger simulations (here we only have S=100) or larger samples, Lin's estimator consistently hits the 95% target.

4. Lin's estimator has the lowest variance at 0.188. Lin's estimator is therefore the most efficient, producing the most precise ATE estimates among the three versions. …

Monte Carlo 5, Bonus: Selection Bias

Imagine a scenario where the probability of treatment depends on X.

- $X \in \{0, 1\}$ with $P(X = 1) = 0.5$.
- Stratified Assignment:
  - If $X = 0$, $P(D = 1) = 0.2$.
  - If $X = 1$, $P(D = 1) = 0.8$.
- Outcome: $Y = 0 + 1 \cdot D + 5 \cdot X + 2 \cdot D \cdot X + \epsilon$.

So $X = 1$ makes you more likely to be treated AND increases your outcome. Treatment effect is also larger for $X = 1$.

Questions:

1. Write a function `dgp_selection(n)` for this design.
2. Run a simulation ($S = 1000, n = 200$) comparing:
   - DIM: `lm(Y ~ D)`
   - Lin: `lm_lin(Y ~ D, covariates = ~ X)`
3. Show that DIM is biased. Calculate the bias.
4. Show that Lin recovers the true ATE.
5. Why does DIM fail here?

Answer

```r
library(tidyverse)
library(gt)
library(gtsummary)
library(estimatr)
library(broom)
```

1. Firstly, we compute the true ATE. The treatment effect is $\tau(X) = 1 + 2X$, so $\tau(0) = 1$ and $\tau(1) = 3$. Since $P(X = 1) = 0.5$, the true ATE is $0.5 \times 1 + 0.5 \times 3 = 2$.

```r
dgp_selection <- function(n) {
  X <- rbinom(n, 1, 0.5)
  prob_treat <- ifelse(X == 0, 0.2, 0.8)
  D <- rbinom(n, 1, prob_treat)
  e <- rnorm(n, 0, 1)
  Y <- 0 + 1 * D + 5 * X + 2 * D * X + e
  return(tibble(D = D, X = X, Y = Y))
}
```

2. Now that we have dgp, we build a simulation function comparing DIM and Lin. And run it $S = 1000$ times with $n = 200$.

```r
sim_selection <- function(s, n) {
  df <- dgp_selection(n)
  thrsh <- 2 # The ATE we calcualted, thrsh- short for 'threshold'

  fit_dim <- lm(Y ~ D, data = df)
  res_dim <- tidy(fit_dim, conf.int = TRUE) |>
           filter(term == "D") |>
           mutate(estimator = "DIM", covered = (conf.low < thrsh & conf.high >
thrsh))

  fit_lin <- lm_lin(Y ~ D, covariates = ~ X, data = df)
  res_lin <- tidy(fit_lin, conf.int = TRUE) |>
           filter(term == "D") |>
           mutate(estimator = "Lin", covered = (conf.low < thrsh & conf.high >
thrsh))

  bind_rows(res_dim, res_lin) |>
     mutate(s = s)
}
```

```r
S <- 1000
n <- 200

set.seed(69)

sim_results <- 1:S |>
  map_df(sim_selection, n = n)

summary_results <- sim_results |>
  group_by(estimator) |>
  summarise(
    mean_ATE_hat = mean(estimate),
    sd_ATE_hat = sd(estimate),
    coverage = mean(covered),
    .groups = "drop"
  )

summary_results
```

```
# A tibble: 2 × 4
  estimator mean_ATE_hat sd_ATE_hat coverage
  <chr>            <dbl>      <dbl>    <dbl>
1 DIM               5.60      0.386    0
2 Lin               2.01      0.203    0.923
```

3. We calculate the bias by comparing mean estimates to the true ATE of 2.

```
true_ATE <- 2

summary_results |>
  mutate(
    bias = mean_ATE_hat - true_ATE,
    bias_pct = (bias / true_ATE) * 100
  ) |>
  select(estimator, mean_ATE_hat, bias, bias_pct)
```

```
# A tibble: 2 × 4
  estimator mean_ATE_hat   bias bias_pct
  <chr>            <dbl>  <dbl>    <dbl>
1 DIM               5.60 3.60    180.
2 Lin               2.01 0.0109    0.545
```

The DIM estimator is severely biased upward—approximately 70% bias. DIM overestimates the treatment effect because it conflates the effect of X with the treatment effect.

4. Lin's mean estimate is approximately 2(1.99), matching the true ATE. The bias is essentially zero, and coverage is close to 95%. Lin successfully recovers the true ATE.

5. DIM fails because of selection/omitted variable bias. X affects both treatment assignment ($P(D = 1 \mid X = 1) = 0.8$ vs $P(D = 1 \mid X = 0) = 0.2$) and the outcome. When we ignore X, the treated group is mostly high-X individuals (higher Y) while the control group is mostly low-X individuals (lower Y). DIM attributes to treatment what is actually due to X. while lin succeeds by controlling for X and allowing for treatment effect heterogeneity.

…

Linear regression for experiments

Job corps, reanalysis

This question will be demonstrated in class.

We will use the same data as in the "Job corps" question in Assignment 1:

```r
library(causalweight)
```

```
Loading required package: ranger
```

```r
data(JC)
JC <- as_tibble(JC)
```

Here is one way to compute the difference in means of the treatment and control group. compare it to the result on H, p. 21.

```r
JC |> filter(assignment == 1) -> JC_short_TG
JC |> filter(assignment == 0) -> JC_short_CG
treat_mean <- mean(JC_short_TG$earny4)
control_mean <- mean(JC_short_CG$earny4)
(ATE_hat <- treat_mean - control_mean)
```

```
[1] 16.05513
```

That is identical to the estimated effect in H3.1.

Questions:

1. Estimate the ATE using `lm_robust` and present the results in a regression table using `tbl_regression`.
2. Run a regression that reveals whether the effect is different for men and women. Use `lm_lin` with the `covariates` argument.

Interpret all your results.

Answer

1. Estimate the ATE using `lm_robust` and present the results in a regression table using `tbl_regression`.

```r
library(estimatr)
library(gtsummary)

ate_fit <- lm_robust(
  earny4 ~ assignment,
  data = JC
)

tbl_regression(
  ate_fit,
  estimate_fun = ~style_sigfig(.x, digits = 3)
)
```

Characteristic	**Beta**	**95% CI**	**p-value**
assignment	16.1	8.07, 24.0	<0.001

Abbreviation: CI = Confidence Interval

Step 1 (Load Packages): Loading library(estimatr), loaded "lm_robust", the function that we used to determine the estimator. Loading "gtsummary" puts the regression outputs into tables. This package was used when we ran "tbl_regression.

Step 2 (store regression results in object called ate_fit): The second step was to use lm_robust() to run a regression. This package uses heteroskedasticity-robust standard errors. Within this function, we regressed the assignment (treatment), onto earny4 (the outcome). The line "data = JC", indicates to R to use the data from the JC dataset.

Step 3 (express results in a table): Our last step was to put these results into a tabular form. style_sigfig(), indicates for numbers to be formatted using significant figures. We told R to show 3 significant digits. The x in the expression is a placeholder.

The beta coefficient on assignment is the DIM of fourth-year earnings between individuals randomly assigned to Job Corps (treatment) compared to those who were not assigned to JC (the control). The coefficient equals 16.1 which is close to the naive DIM calculated in Huber, pg.21. Since the treatment is random, we can state that 16 suggests a casual effect of the JC program.

2. Run a regression that reveals whether the effect is different for men and women. Use `lm_lin` with the `covariates` argument.

```
lin_gender <- lm_lin(
  earny4 ~ assignment,
  covariates = ~ female,
  data = JC
)

tbl_regression(
  lin_gender,
  estimate_fun = ~style_sigfig(.x, digits = 3)
)
```

```
✖ Unable to identify the list of variables.
```

```
This is usually due to an error calling `stats::model.frame(x)`or
`stats::model.matrix(x)`.
It could be the case if that type of model does not implement these methods.
Rarely, this error may occur if the model object was created within
a functional programming framework (e.g. using `lappy()`, `purrr::map()`,
etc.).
```

Characteristic	**Beta**	**95% CI**	**p-value**
(Intercept)	195	189, 201	<0.001
assignment	21.0	13.1, 28.9	<0.001
female_c	−64.0	−75.8, −52.2	<0.001
assignment:female_c	−5.16	−20.8, 10.4	0.5

Abbreviation: CI = Confidence Interval

Step 1: We ran Lin's estimator and stored the result into a model object named "lin_gender". We first defined the outcome was earnings in year 4, and the treatment assignment. This denotes our standard regression for the ATE.

Lin's estimator is a function in R that uses the methods from Lin (2013). It is used when estimating treatment effects with pre-treatment covariate data, in this example the pre-treatment covariate data was gender. This method processes the data using the covariates specified in the `covariates` argument (female). It then centers them by subtracting from each covariate its mean, and interacting them with the treatment. This is a form of regression adjustment. This is a useful function because it allows for a strong flexible form in the regression.

Lastly, we indicate the dataset (JC Corps), using data=JC.

Step 2: We put the output into a table and express how many figures are to be displayed.

We can interpret the results of the table as with the following description:

Job Corps increases year-4 earnings by about 21 at the sample mean of gender.

Women have lower earnings than men in the control group (about 64 lower, relative to the centered baseline).

The treatment effect for women is about 5.16 smaller than for men, this is displayed in the interaction term.

HIV RCT, reanalysis

We will analyze the effect of the HIV information campaign in Thornton (2008).

After you complete the data analysis below, please also answer the following substantive questions:

1. What exactly is the specific policy intervention studied in this paper?
2. Can you think of alternative policy interventions that might achieve similar goals?
3. Based on the results, is this an effective policy? What would you recommend to the health authority in Malawi?
4. Would you recommend this policy to the Canadian health authorities?
5. What is your main criticism of Thornton's study?

We first need to conduct the empirical analysis. Use the same steps as in the previous exercise to analyze the effect of the campaign (`any`). Use `lm_robust` and `tbl_regression`. Explore whether the treatment effect differs by age (heterogeneity), by creating a binary variable for age (e.g. median split or `age >= 33`) and using `lm_lin`.

Remember to use `drop_na` after loading the data to remove the missing data.

Answer

First we load the data and remove missing values.

```r
install.packages("causaldata")
library(estimatr)
library(gtsummary)
```

```r
library(causaldata)
data("thornton_hiv")
Thornton <- thornton_hiv |>
drop_na() |>
as_tibble()
```

Next step was to estimate the ATE of the incentive (`any`) on obtaining results (`got`), robustly.

```r
ate_model <- lm_robust(
  got ~ any,
  data = Thornton
)

tbl_regression(
  ate_model
)
```

Characteristic	**Beta**	**95% CI**	**p-value**
Received any incentive	0.45	0.41, 0.49	<0.001

Abbreviation: CI = Confidence Interval

The coefficient on any is positive and statistically significant(CI: .41-.49), indicating that monetary incentives increased the probability of individuals learning their HIV results.

Now we use `lm_lin`. We create a binary age variable using a median split, to explore heterogeneity.

```r
median_age <- median(Thornton$age)

Thornton <- Thornton |>
  mutate(age_above_median = ifelse(age >= median_age, 1, 0))
```

```r
ate_model <- lm_robust(
  got ~ any * age_above_median,
  data = Thornton
)

tbl_regression(ate_model)
```

Characteristic	**Beta**	**95% CI**	**p-value**
Received any incentive	0.48	0.42, 0.53	<0.001
age_above_median	0.10	0.02, 0.17	0.010
Received any incentive * age_above_median	−0.06	−0.14, 0.03	0.2

Abbreviation: CI = Confidence Interval

The interaction term shows whether the treatment effect differs for older vs younger individuals.

Answers:

1. The intervention is the randomized monetary incentive to encourage individuals in Malawi to collect their HIV test results. Participants were randomized to receive varying amounts of money (or nothing) for returing to the clinic to learn their results.
2. If the incentive is to encourage individuals to observe/collect their results then the fricton is the effort/ability to go to the testing clininc. Alternatives could be mailed results, SMS results (however I'm unsure how widespread cellular use was in Malawi in 2008).
3. The policy is effective—the ATE is positive and significant. As such we would recommend implementing the programs in areas with low uptake.
4. Recommendation for Canada? No, now this answer is explicity an opinion and not necessary fact. In Canada, frictions are more about stigma and awareness than poverty. Monetary incen-

tives may be less cost-effective and could reinforce stigma. It could be more efficent to invest in self-testing kits, and targeted outreach to high-risk populations.

5. Our main critisicm is that the external validity is limited (rural Malawi context). The study only measures short-term result collection, not longer-term outcomes like behavior change or treatment initiation. The sample already agreed to be tested, so results may not generalize to convincing untested populations.

…

Project STAR

When loading libraries, we sometimes want to suppress messages and warnings to keep our document clean. We can do this by setting `#| message: false` and `#| warning: false` in the chunk options.

Load the `AER` library (install first, if necessary), and the `STAR` data that comes with it:

```r
library(AER)
data("STAR")
STAR <- as_tibble(STAR) |> drop_na()
```

From `?STAR` (which you should look at, to find the variable descriptions):

> Project STAR (Student/Teacher Achievement Ratio) was a four-year longitudinal class-size study funded by the Tennessee General Assembly and conducted in the late 1980s by the State Department of Education. Over 7,000 students in 79 schools were randomly assigned into one of three interventions: small class (13 to 17 students per teacher), regular class (22 to 25 students per teacher), and regular-with-aide class (22 to 25 students with a full-time teacher's aide). Classroom teachers were also randomly assigned to the classes they would teach. The interventions were initiated as the students entered school in kindergarten and continued through third grade.

We will focus on 3rd grade effects. We summarize the treatment variable using `tbl_summary` from the `gtsummary` package for a nicer look:

```r
STAR |>
    select(star3) |>
    tbl_summary()
```

Characteristic	**N = 2,161**[1]
star3	
regular	548 (25%)
small	858 (40%)
regular+aide	755 (35%)

[1] n (%)

We first define the treatment variable as 0 if class is "regular", 1 if small:

```r
STAR |>
    filter(star3 %in% c("regular", "small")) |>
    mutate(D3 = ifelse(star3 == "small", 1, 0)) -> STAR_binary
```

First, compute the treatment effect of `D3` on total reading scaled score in 3rd grade. Make sure to use heteroskedasticity-robust standard errors throughout. Interpret the results. Second, compute the effect on total math scaled score in 3rd grade. Interpret the results.

Third, run the regression for math using "multiple treatments", so using `star3` as the treatment variable. Make sure to use `STAR`, not `STAR_binary`. Do you find the same results for the `small` treatment? Interpret your results for the `regular+aide` treatment.

Bonus points: can you find evidence of heterogeneity of the treatment effect?

Answer

Checking the variables of the data set

```
?STAR
```

1. Using 'lm_robust' (which builds heteroskedasticity-robust inference in by default), we will compute the treatment effect of 'D3' on the total reading scaled score in the 3rd grade, 'read3'.

```
library(estimatr)

read3_mod_robust <- lm_robust(
  read3 ~ D3,
  data = STAR_binary,
  se_type = "HC1"   # heteroskedasticity-robust SEs
)

summary(read3_mod_robust)
```

```

Call:
lm_robust(formula = read3 ~ D3, data = STAR_binary, se_type = "HC1")

Standard error type:  HC1

Coefficients:
            Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper   DF
(Intercept)  624.429      1.575 396.552  0.00000  621.340  627.518 1404
D3             5.107      2.029   2.518  0.01193    1.128    9.087 1404

Multiple R-squared:  0.004462 , Adjusted R-squared:  0.003753
F-statistic: 6.338 on 1 and 1404 DF,  p-value: 0.01193
```

The coefficient on D3 is 5.107 and the p-value is less than 0.05. Therefore, using heteroskedasticity-robust standard errors, assignment to a small class increases 3rd-grade reading scores by about 5 points relative to regular classes, and the effect is statistically significant at the 5% level.

2. Next, we will compute the effect on total math scale scored in the 3rd grade, 'math3'. This is very similar to the code above.

```r
math3_mod <- lm_robust(
  math3 ~ D3,
  data = STAR_binary,
  se_type = "HC1"
)

summary(math3_mod)
```

```

Call:
lm_robust(formula = math3 ~ D3, data = STAR_binary, se_type = "HC1")

Standard error type:  HC1

Coefficients:
            Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper   DF
(Intercept)  627.261      1.660 377.982   0.0000  624.006  630.516 1404
D3             3.075      2.145   1.434   0.1519   -1.133    7.282 1404

Multiple R-squared:  0.001446 , Adjusted R-squared:  0.0007349
F-statistic: 2.055 on 1 and 1404 DF,  p-value: 0.1519
```

The coefficient on D3 is 3.075, but the p-value is high at 0.1519. Assignment to a small class therefore increases 3rd-grade math scores by about 3 points, but the effect is not statistically significant, with a 95% confidence interval that includes zero.

3. Lastly, we will run the regression for math using multiple treatments with `star3` as the treatment variable. Each treatment is encoded as a binary indicator (R handles factor variables as binary in regression by default). The intercept corresponds to the omitted (baseline) category ('regular') and each coefficient compares its group to the reference group.

```r
math3_multi <- lm_robust(
  math3 ~ star3,
  data = STAR,
  se_type = "HC1"
)

summary(math3_multi)
```

```

Call:
lm_robust(formula = math3 ~ star3, data = STAR, se_type = "HC1")
```

```
Standard error type:  HC1

Coefficients:
                  Estimate Std. Error  t value Pr(>|t|) CI Lower CI Upper   DF
(Intercept)        627.261      1.659 377.9882   0.0000  624.007  630.515 2158
star3small           3.075      2.145   1.4336   0.1518   -1.131    7.281 2158
star3regular+aide    1.327      2.196   0.6043   0.5457   -2.979    5.634 2158

Multiple R-squared:  0.0009839 ,    Adjusted R-squared:  5.805e-05
F-statistic: 1.069 on 2 and 2158 DF,  p-value: 0.3435
```

The effect of being in a small class is positive (3.075 points) and unchanged whether we use the binary or multiple-treatment specification, and the standard error is the same. This is because 'regular' remains the omitted reference category in both models.

The effect of being in a regular class with an aide is small (around 1.3 points) and also not statistically significant, indicating that there is no evidence that adding a full-time aide to a regular-sized class improves 3rd-grade math scores.

Bonus: we can explore whether the effect of small class sizes varies across subgroups (hetergenous treatment effects). For example, we can see whether the effect of small class sizes differs depending on student characteristics like gender, SES or teachers characteristics like experience teaching, etc.

We'll start by observing the effect across gender (again, keeping our focus to the thrid grade reading and math scores).

```r
# Heterogeneity by gender - reading scores
read3_gender_mod <- lm_robust(
  read3 ~ D3 * gender,
  data = STAR_binary,
  se_type = "HC1"
)

summary(read3_gender_mod)
```

```
Call:
lm_robust(formula = read3 ~ D3 * gender, data = STAR_binary,
    se_type = "HC1")

Standard error type:  HC1

Coefficients:
                Estimate Std. Error   t value Pr(>|t|) CI Lower CI Upper   DF
(Intercept)     622.2530      2.275 273.51128  0.00000 617.7901  626.716 1402
D3                5.2374      2.939   1.78217  0.07494  -0.5275   11.002 1402
```

```
genderfemale      4.0420      3.148   1.28386  0.19940  -2.1339   10.218 1402
D3:genderfemale  -0.0708      4.057  -0.01745  0.98608  -8.0302    7.889 1402

Multiple R-squared:  0.007329 ,	Adjusted R-squared:  0.005205 
F-statistic: 3.568 on 3 and 1402 DF,  p-value: 0.01367
```

```
# Heterogeneity by gender - math score
math3_gender_mod <- lm_robust(
  math3 ~ D3 * gender,
  data = STAR_binary,
  se_type = "HC1"
)

summary(read3_gender_mod)
```

```

Call:
lm_robust(formula = read3 ~ D3 * gender, data = STAR_binary, 
    se_type = "HC1")

Standard error type:  HC1 

Coefficients:
                Estimate Std. Error   t value Pr(>|t|) CI Lower CI Upper   DF
(Intercept)     622.2530      2.275 273.51128  0.00000 617.7901  626.716 1402
D3                5.2374      2.939   1.78217  0.07494  -0.5275   11.002 1402
genderfemale      4.0420      3.148   1.28386  0.19940  -2.1339   10.218 1402
D3:genderfemale  -0.0708      4.057  -0.01745  0.98608  -8.0302    7.889 1402

Multiple R-squared:  0.007329 ,	Adjusted R-squared:  0.005205 
F-statistic: 3.568 on 3 and 1402 DF,  p-value: 0.01367
```

```
summary(math3_gender_mod)
```

```

Call:
lm_robust(formula = math3 ~ D3 * gender, data = STAR_binary, 
    se_type = "HC1")

Standard error type:  HC1 

Coefficients:
                Estimate Std. Error  t value Pr(>|t|) CI Lower CI Upper   DF
(Intercept)      627.842      2.533 247.8242   0.0000  622.872  632.812 1402
```

```
D3                  4.538      3.180   1.4268   0.1539   -1.701   10.777 1402
genderfemale       -1.079      3.348  -0.3223   0.7473   -7.648    5.489 1402
D3:genderfemale    -2.889      4.311  -0.6701   0.5029  -11.345    5.568 1402

Multiple R-squared:  0.003062 , Adjusted R-squared:  0.0009289
F-statistic: 1.475 on 3 and 1402 DF,  p-value: 0.2196
```

There is no evidence of heterogeneity of the small-class effect by gender: small classes increase reading scores by about 5 points for both boys and girls, and the difference in treatment effects across gender is essentially zero and not statistically significant. Additionally, the difference in treatment effects across gender for math scores is negative but not statistically significant.

Next, we'll try by free lunch status, an indicator of socio-economic status (SES).

```r
# Testing if the effect differs by Free Lunch status (SES) - reading
model_het <- lm_robust(read3 ~ D3 * lunch3, data = STAR_binary)
summary(model_het)
```

```


Call:
lm_robust(formula = read3 ~ D3 * lunch3, data = STAR_binary)

Standard error type:  HC2

Coefficients:
              Estimate Std. Error  t value  Pr(>|t|) CI Lower CI Upper   DF
(Intercept)    631.256      1.835 344.0555 0.000e+00  627.656  634.855 1402
D3               5.662      2.394   2.3657 1.813e-02    0.967   10.358 1402
lunch3free     -19.485      3.279  -5.9429 3.528e-09  -25.916  -13.053 1402
D3:lunch3free   -0.946      4.173  -0.2267 8.207e-01   -9.131    7.239 1402

Multiple R-squared:  0.07096 ,  Adjusted R-squared:  0.06897
F-statistic: 34.43 on 3 and 1402 DF,  p-value: < 2.2e-16
```

```r
# Testing if the effect differs by Free Lunch status (SES) - math
model_het <- lm_robust(math3 ~ D3 * lunch3, data = STAR_binary)
summary(model_het)
```

```


Call:
lm_robust(formula = math3 ~ D3 * lunch3, data = STAR_binary)

Standard error type:  HC2

Coefficients:
```

```
              Estimate Std. Error  t value  Pr(>|t|) CI Lower CI Upper   DF
(Intercept)    634.022      1.938 327.0709 0.000e+00 630.2198  637.825 1402
D3               4.490      2.570   1.7475 8.077e-02  -0.5503    9.531 1402
lunch3free     -19.299      3.473  -5.5570 3.281e-08 -26.1111  -12.486 1402
D3:lunch3free   -3.334      4.372  -0.7625 4.459e-01 -11.9096    5.242 1402

Multiple R-squared:  0.06909 ,  Adjusted R-squared:  0.06709
F-statistic: 34.65 on 3 and 1402 DF,  p-value: < 2.2e-16
```

Here, the interaction is nearly zero, and not statistically significant. This suggests that while free lunch status is a huge predictor of reading scores (the −19.48 drop), it doesn't actually change how effective the small classroom size is. Math scores for the third grade have a negative effect, again not statistically significant.

Lastly we'll try by ethnicity using lm_lin.

```r
# Using lm_lin for the ethnicity heterogeneity check
model_lin_eth <- lm_lin(
  formula = read3 ~ D3,
  covariates = ~ ethnicity,
  data = STAR_binary
)

summary(model_lin_eth)
```

```
1 coefficient  not defined because the design matrix is rank deficient
```

```


Call:
lm_lin(formula = read3 ~ D3, covariates = ~ethnicity, data = STAR_binary)

Standard error type:  HC2

Coefficients: (1 not defined because the design matrix is rank deficient)
                       Estimate Std. Error  t value  Pr(>|t|) CI Lower CI
Upper
(Intercept)             624.253      1.526 409.0640 0.000e+00  621.259
627.246
D3                        5.321      1.970   2.7010 6.997e-03    1.456
9.185
ethnicityafam_c         -21.834      3.426  -6.3740 2.495e-10  -28.554
-15.114
ethnicityasian_c          3.705      6.928   0.5348 5.929e-01   -9.885
17.296
ethnicityhispanic_c          NA         NA       NA        NA       NA
NA
```

```
D3:ethnicityafam_c        2.631      4.464   0.5894 5.557e-01   -6.127
11.389
D3:ethnicityasian_c     -28.061     10.682  -2.6269 8.710e-03  -49.016
-7.107
D3:ethnicityhispanic_c   30.644     15.070   2.0334 4.220e-02    1.082
60.207
                         DF
(Intercept)            1399
D3                     1399
ethnicityafam_c        1399
ethnicityasian_c       1399
ethnicityhispanic_c      NA
D3:ethnicityafam_c     1399
D3:ethnicityasian_c    1399
D3:ethnicityhispanic_c 1399

Multiple R-squared:  0.06097 ,  Adjusted R-squared:  0.05694
F-statistic:    NA on 6 and 1399 DF,  p-value: NA
```

The coefficient in front of the interaction term for treatment and asian ethnicity is -28, a statistically significant and large negative effect. It suggests the treatment was much less effective on reading scores for the Asian students in this specific sample.

While there is evidence of heterogeneity—specifically a lower observed effect for Asian students (p=0.008)—small subgroup sizes for certain ethnicities (notably Hispanic students) lead to rank deficiency, suggesting that subgroup-specific results should be interpreted with caution.

For troubleshooting: do not edit or remove

```
sysname

"Darwin"

release

"24.6.0"

version
"Darwin Kernel Version 24.6.0: Wed Nov  5 21:34:00 PST 2025;
root:xnu-11417.140.69.705.2~1/RELEASE_ARM64_T8132"

nodename

"Jenetas-MacBook-Air.local"

machine

"arm64"

login

"root"

user

"jenetaljutic"

effective_user

"jenetaljutic"
```

```
[1] "2026-02-02 10:16:59 EST"
```