Problem Set 1: Potential Outcomes and Randomised Experiments

MY457 | WT 2026 | Week 2

Add your answers to this .qmd file by replacing the dedicated placeholder text. When you are finished, render the .qmd file as a .pdf and make sure you push your work to GitHub.

1 Concepts

This question reviews some of the concepts covered in class. Mathematical notation is often necessary to be specific about the assumptions and quantities of interest. If you want to support your explanations with mathematical notation, this page provides a tutorial on including mathematical notation in Quarto.

Consider a simple study of the effect of a treatment $D_i \in \{0, 1\}$ on Y_i for all units $i \in \{1, 2, 3, \dots, N\}$.

1.1 In the table below, fill in the missing values. For one unit, this is not possible. Which one is it, and why?

i	D_i	Y_{0i}	Y_{1i}	Y_i
1	0	5	10	5
2	1	3	8	**?**
3	**?**	4	4	4
4	0	**?**	12	7
5	**?**	4	9	9

Solution

To solve this task, you can use the switching equation (Equation 1) and solve for the missing values.

$$Y_i = Y_{0i} \cdot (1 - D_i) + Y_{1i} \cdot D_i \tag{1}$$

For instance, consider unit 2:

- $D_2 = 1$
- $Y_{0,2} = 3$
- $Y_{1,2} = 8$

Using the switching equation, we get:

$$\begin{aligned} Y_i &= 3 \cdot (1 - 1) + 8 \cdot 1 \\ Y_i &= 3 \cdot 0 + 8 \\ Y_i &= 8 \end{aligned}$$

Intuitively, since unit 2 received the treatment ($D_2 = 1$), their observed outcome Y_2 is equal to the potential outcome under treatment $Y_{1,2}$, which is 8.

i	D_i	Y_{0i}	Y_{1i}	Y_i
1	0	5	10	5
2	1	3	8	**8**
3	**?**	4	4	4
4	0	**7**	12	7
5	**1**	4	9	9

Unit 3 is the one for which we cannot determine the missing value. This is because for unit 3, both potential outcomes are identical ($Y_{0i} = Y_{1i} = 4$). Thus, we cannot infer whether the unit received treatment or not based on the observed and the potential outcomes.

1.2 What is the individual treatment effect τ_i for unit $i = 1$ in the table above? Explain how you arrived at your answer.

Solution

The individual treatment effect τ_i for unit i is defined as the difference between the potential outcome under treatment and the potential outcome under control: $\tau_i = Y_{1i} - Y_{0i}$.
Hence, for unit 1:

$$\begin{aligned}\tau_1 &= Y_{1,1} - Y_{0,1} \\ &= 10 - 5 \\ &= 5\end{aligned}$$

1.3 Given the table of potential outcomes below, calculate the Average Treatment Effect (ATE), the Average Treatment Effect on the Treated (ATT), and the Average Treatment Effect on the Untreated (ATU).

D_i	Y_{0i}	Y_{1i}
0	6	10
0	4	8
1	5	12
1	7	14

Solution

The ATE is the average of the individual treatment effects across all units:

$$\begin{aligned}\tau_{ATE} &= \frac{1}{N}\sum_{i=1}^{N}(Y_{1i} - Y_{0i}) \\ &= \frac{(10-6) + (8-4) + (12-5) + (14-7)}{4} \\ &= \frac{4+4+7+7}{4} \\ &= \frac{22}{4} \\ &= 5.5\end{aligned}$$

The ATT is the average of the individual treatment effects across units that received the

treatment ($D_i = 1$):

$$\begin{aligned}
\tau_{ATT} &= \frac{1}{N_1} \sum_{i=1}^{N} D_i(Y_{1i} - Y_{0i}) \\
&= \frac{(12-5) + (14-7)}{2} \\
&= \frac{7+7}{2} \\
&= \frac{14}{2} \\
&= 7
\end{aligned}$$

The ATU is the average of the individual treatment effects across units that did not receive the treatment ($D_i = 0$):

$$\begin{aligned}
\tau_{ATU} &= \frac{1}{N_0} \sum_{i=1}^{N} (1 - D_i)(Y_{1i} - Y_{0i}) \\
&= \frac{(10-6) + (8-4)}{2} \\
&= \frac{4+4}{2} \\
&= \frac{8}{2} \\
&= 4
\end{aligned}$$

1.4 Compare the following two quantities: $E[Y_i \mid D_i = 1] - E[Y_i \mid D_i = 0]$ and $E[Y_{1i} - Y_{0i}]$. What does each represent, and under what conditions are they equal?

Solution

The quantity $E[Y_i \mid D_i = 1] - E[Y_i \mid D_i = 0]$ is the difference in average observed outcomes between the treated and untreated groups. This is often called the "difference in means estimator" or the "naive estimator" of the treatment effect.

In contrast, $E[Y_{1i} - Y_{0i}]$ is the Average Treatment Effect (ATE), which is the average difference in potential outcomes for all units in the population. This is a theoretical quantity (an estimand) and not directly observable.

The difference in means estimator is an unbiased estimator of the ATE if treatment assignment is independent of the potential outcomes—that is, if $D_i \perp (Y_{0i}, Y_{1i})$. While

we can never test whether this independence assumption holds (we'd have to observe both potential outcomes to do so), random assignment of D - in expectation - leads to independence of potential outcomes and treatment assignment. Hence, when random assignment is used, the treated and untreated groups are - in expectation - comparable in terms of their potential outcomes, so the difference in observed means is an unbiased estimator of the ATE.
Note that random assignment ensures balanced potential outcomes only **in expectation**; in any given sample, there may still be differences due to random variation. Yet, if we repeated the random assignment process many times, the average of the difference in means across all these repetitions would converge to the ATE.

2 Simulation

In this section, we will use simulated data to test some of our intuitions about randomised experiments. The advantage of using a simulated dataset is that we have explicit control over the data generating process, and know the 'true' answer to any question we pose.

We will first simulate a dataset that corresponds to the table from above:

```
dat1 <- tibble(
    D = c(0, 0, 1, 1),
    Y0 = c(6, 4, 5, 7),
    Y1 = c(10, 8, 12, 14),
    Y = Y0 * (1 - D) + Y1 * D
)
```

2.1 Using the simulated dataset above, calculate the difference-in-means and compare it to the true ATE. Are they equal? Why or why not?

Solution

```
# Calculate the difference in means
diff_in_means <- mean(dat1$Y[dat1$D == 1]) - mean(dat1$Y[dat1$D == 0])
# Calculate the true ATE
true_ATE <- mean(dat1$Y1 - dat1$Y0)
```

No, they are not equal. The true ATE is 5.5, while the difference in means estimator yields 8. The two quantities would be equal in expectation if `D` was randomly assigned. We don't know whether that's the case in the above example, because we don't know the data generation process.

However, even if D was randomly assigned, we only observe four units, so random variation would most likely still lead to differences between the two quantities. Recall that the difference-in-means estimator is an unbiased estimator of the ATE **in expectation**. With random assignment the difference-in-means estimator would converge to the true ATE over many repeated samples or with a large sample size.

2.2 Explain what the simulation code below does. Given the data generating process, why can't we use the naive difference-in-means estimator to recover the true ATE? How would you modify the data generating process to ensure that the naive difference-in-means estimator can be used to recover the true ATE?

```
n <- 5000
tau <- 5

# always set a seed for reproducibility
set.seed(2026)
dat2 <- tibble(
    Y0 = rnorm(n, mean = 50, sd = 10),
    Y1 = Y0 + tau,
    D = rbinom(n, size = 1, prob = 0.01 * Y0),
    Y = Y0 * (1 - D) + Y1 * D
)
```

Solution

The simulation code generates a dataset with 5000 units. The potential outcome under control ($Y0$) is drawn from a normal distribution with a mean of 50 and a standard deviation of 10. The potential outcome under treatment ($Y1$) is defined as $Y0+\tau$, where τ is set to 5, indicating a constant treatment effect across all units.
The treatment assignment (D) is generated using a binomial distribution, where the probability of receiving treatment is dependent on the value of $Y0$ (specifically, $0.01*Y0$). This means that units with higher values of $Y0$ are more likely to receive the treatment. This is an example of selection bias, as the treatment assignment is not independent of the potential outcomes. Hence, **we cannot use the naive difference in means estimator to recover the true ATE**.
To ensure that we can use the naive difference in means estimator to recover the true ATE, we need to change the way treatment is assigned. Instead of making the probability of treatment dependent on $Y0$, we should assign treatment purely at random. Typically, we'd use a probability of 0.5 for random assignment:

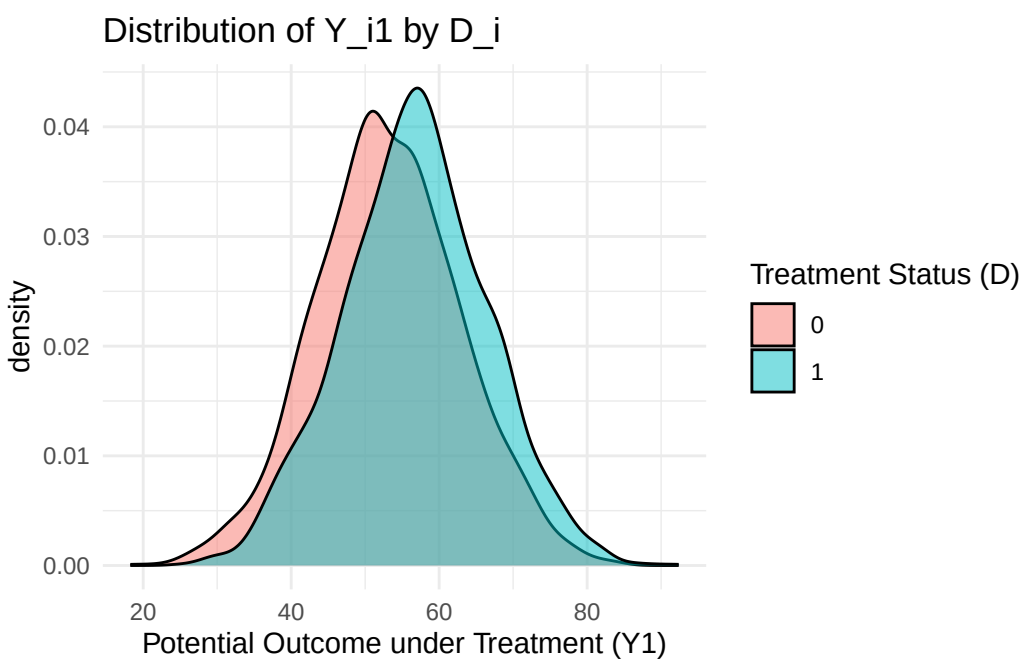
```r
D <- rbinom(n, size = 1, prob = 0.5)
```

2.3 Create a plot that visualises the distribution of the potential outcomes under treatment (Y_{1i}) by treatment status D_i using the simulated dataset `dat2` from above. What do you observe?

Solution

Since the treatment assignment is dependent on the potential outcomes under control, the simulated potential outcomes under treatment ($Y1$) differ systematically between the treated and untreated group. Specifically, units with higher $Y0$ values are more likely to receive treatment, leading to a higher average $Y1$ in the treated group compared to the untreated group.

```r
library(ggplot2)
ggplot(dat2, aes(x = Y1, fill = factor(D))) +
    geom_density(alpha = 0.5) +
    labs(
        title = "Distribution of Y_i1 by D_i",
        x = "Potential Outcome under Treatment (Y1)",
        fill = "Treatment Status (D)"
    ) +
    theme_minimal()
```

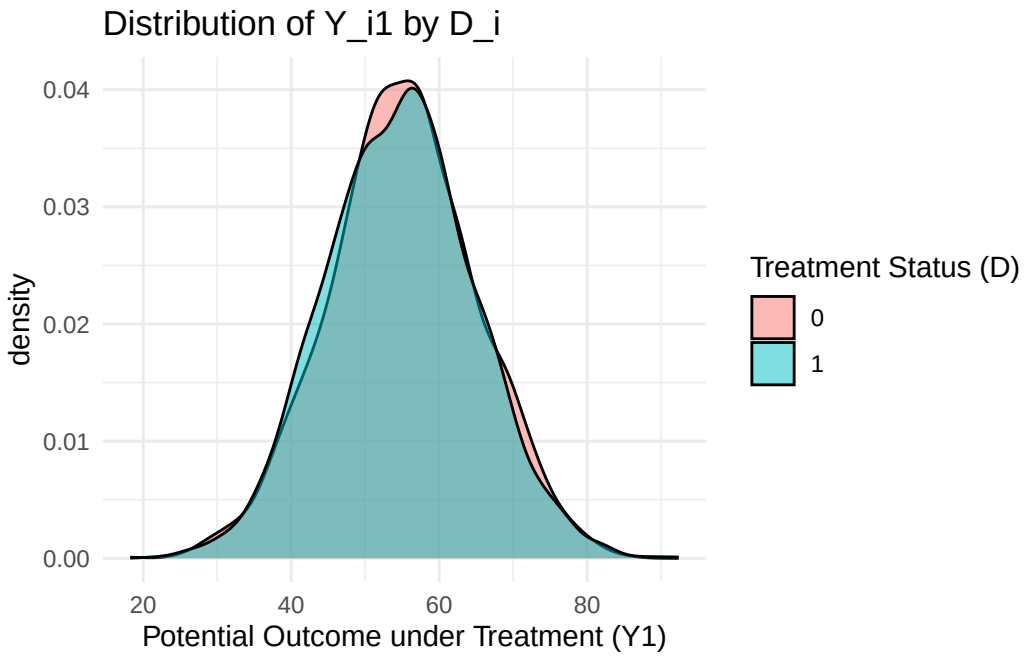


2.4 Next, create a new dataframe `dat3` which includes the potential outcomes from `dat2` and a treatment indicator `D` that randomly assigns treatment with probability 0.5. Recalculate the observed outcome `Y` based on this new treatment assignment. Create the same plot as above using the new treatment assignment vector. What do you observe now?

Solution

With the new random treatment assignment, the distribution of potential outcomes under treatment ($Y1$) is almost identical between the treated and untreated groups. This is because treatment is now assigned independently of the potential outcomes, eliminating selection bias. Slight differences may still exist due to random variation but will average out over many repetitions.

```r
set.seed(2026)
dat3 <- tibble(
    D = rbinom(n, size = 1, prob = 0.5),
    Y0 = dat2$Y0,
    Y1 = dat2$Y1,
    # Recalculate observed outcome Y
    Y = Y0 * (1 - D) + Y1 * D
)

# Plotting the distribution of Y1 by treatment status D
ggplot(dat3, aes(x = Y1, fill = factor(D))) +
    geom_density(alpha = 0.5) +
    labs(
        title = "Distribution of Y_i1 by D_i",
        x = "Potential Outcome under Treatment (Y1)",
        fill = "Treatment Status (D)"
    ) +
    theme_minimal()
```



Note: if you regenerated the potential outcomes instead of reusing the ones from `dat2`, you probably got slightly different results, even if you used the same seed. That is because the seed affects the `rbinom` code differently depending on whether there is an `rnorm()` call before or not:

```r
set.seed(123)
rbinom(3, 1, .5)
```

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

```r
set.seed(123)
rnorm(1)
```

```
[1] -0.5604756
```

```r
rbinom(3, 1, .5)
```

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

2.5 Using the data in `dat3`, estimate the ATE using both linear regression and the difference-in-means estimator. Do the estimates produced by these estimators differ? How well do they recover the estimand?

Solution

```r
# Estimate the difference in means using linear regression
ate_lm <- estimatr::lm_robust(Y ~ D, data = dat3)
ate_dim <- mean(dat3$Y[dat3$D == 1]) - mean(dat3$Y[dat3$D == 0])

# Display results
tidy(ate_lm) |>
    kable(
        digits = 3,
        caption = "ATE Estimate using Linear Regression"
    )
```

Table 4: ATE Estimate using Linear Regression

term	estimate	std.error	statistic	p.value	conf.low	conf.high	df	outcome
(Intercept)	50.111	0.198	252.494	0	49.722	50.500	4998	Y
D	4.618	0.281	16.446	0	4.067	5.168	4998	Y

```
ate_dim
```

```
[1] 4.617567
```

Both estimators yield an estimate of 4.618, which is reasonably close to the estimand (ATE of 5). The fact that we can use linear regression to estimate the difference in means is helpful because it allows us to include covariates and obtain inferential statistics easily.

2.6 For both the `dat2` and `dat3` datasets, calculate the magnitude of selection bias and ATE based on the potential outcomes and estimate the ATE using the difference-in-means estimator (based on the treatment indicator and the observed outcomes). How are the three quantities related? How does the magnitude of selection bias differ between the two datasets, and why?

Solution

```
# For dat2
ATE_dat2 <- mean(dat2$Y[dat2$D == 1]) - mean(dat2$Y[dat2$D == 0])
ATT_dat2 <- mean(dat2$Y1[dat2$D == 1] - dat2$Y0[dat2$D == 1])
selection_bias_dat2 <- mean(dat2$Y0[dat2$D == 1]) - mean(dat2$Y0[dat2$D == 0])

# For dat3
ATE_dat3 <- mean(dat3$Y[dat3$D == 1]) - mean(dat3$Y[dat3$D == 0])
ATT_dat3 <- mean(dat3$Y1[dat3$D == 1] - dat3$Y0[dat3$D == 1])
selection_bias_dat3 <- mean(dat3$Y0[dat3$D == 1]) - mean(dat3$Y0[dat3$D == 0])
```

In `dat2`, selection into treatment is affected by the potential outcomes, leading to a large selection bias of 3.71. In contrast, in `dat3`, treatment is randomly assigned, resulting in a much smaller estimated selection bias of -0.382. However, that the selection bias in dat3 isn't exactly 0 is only due to random variation: the expected value of selection bias in `dat3` under random assignment is 0. With a larger sample size or over multiple simulations, the selection bias in `dat3` would converge to 0.
The ATE we estimate using the difference in means estimator is equal to the sum of the ATT and the selection bias. Because we can't calculate the selection bias from observed data (this is only possible in simulations like this one, where potential outcomes are known), we need research designs that can be expected to yield a selection bias of 0.

```
# The ATE estimated using the difference in means estimator is
# exactly the sum of the ATT and selection bias.
ATE_dat2 == ATT_dat2 + selection_bias_dat2
```

```
[1] TRUE
```

```
ATE_dat3 == ATT_dat3 + selection_bias_dat3
```

```
[1] TRUE
```

2.7 The code below repeatedly simulates a dataset and estimates the ATE and selection bias under random treatment assignment. In a new code chunk, modify the data generating process to introduce selection bias. Run 1000 simulations with your modified process and visualise the distributions of the estimated ATEs and selection biases appropriately. Interpret your findings.

```
n_sims <- 1000
n <- 500
tau <- 2

set.seed(2026)
results <- tibble(
    iter = numeric(n_sims),
    ATE = numeric(n_sims),
    Selection_Bias = numeric(n_sims)
)

# run simulations
for (sim in 1:n_sims) {
    dat <- tibble(
        Y0 = rnorm(n),
        Y1 = Y0 + tau,
        D = rbinom(n, size = 1, prob = 0.5),
        Y = Y0 * (1 - D) + Y1 * D
    )

    # Estimate ATE using difference-in-means
    ate_estimate <- mean(dat$Y[dat$D == 1]) - mean(dat$Y[dat$D == 0])
    results$ATE[sim] <- ate_estimate

    # Calculate Selection Bias
    selection_bias <- mean(dat$Y0[dat$D == 1]) - mean(dat$Y0[dat$D == 0])
    results$Selection_Bias[sim] <- selection_bias
```

```
    # add index
    results$iter[sim] <- sim
}
```

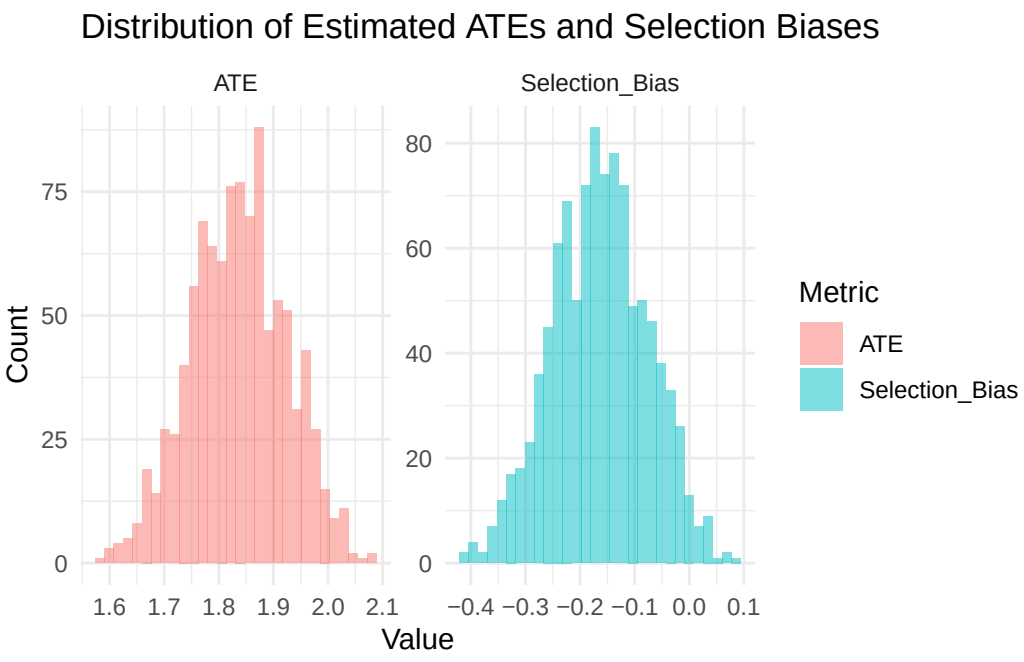
Solution

To introduce selection bias, you have to make the treatment assignment dependent on the potential outcomes. This can take many forms. Here, we introduce a decision rule, where units with $Y0 > 0$ values have a 40% chance of receiving treatment, while units with $Y0 \leq 0$ values have a 50% chance of receiving treatment. This creates a correlation between treatment assignment and potential outcomes, leading to selection bias. We do this by modifying the line that generates `D` in the simulation loop: `D <- rbinom(n, size = 1, prob = ifelse(Y0 > 0, .4, .5))`

```r
n_sims <- 1000
n <- 500
tau <- 2

set.seed(2026)
results <- tibble(
    iter = numeric(n_sims),
    ATE = numeric(n_sims),
    Selection_Bias = numeric(n_sims)
)

# run simulations
for (sim in 1:n_sims) {
    dat <- tibble(
        Y0 = rnorm(n),
        Y1 = Y0 + tau,
        D = rbinom(n, size = 1, prob = ifelse(Y0 > 0, .4, .5)),
        Y = Y0 * (1 - D) + Y1 * D
    )

    # Estimate ATE using difference in means
    ate_estimate <- mean(dat$Y[dat$D == 1]) - mean(dat$Y[dat$D == 0])
    results$ATE[sim] <- ate_estimate

    # Calculate Selection Bias
    selection_bias <- mean(dat$Y0[dat$D == 1]) - mean(dat$Y0[dat$D == 0])
    results$Selection_Bias[sim] <- selection_bias

    # add index
    results$iter[sim] <- sim
}
```

A good way to visualise the results of simulations is to create a histogram. Below, we create histograms of the estimated ATEs and selection biases that show that across many simulations, only a handful of ATE estimates are close to the true value of 2. This is because the selection bias across simulations is not 0 on average, but almost always smaller than 0.

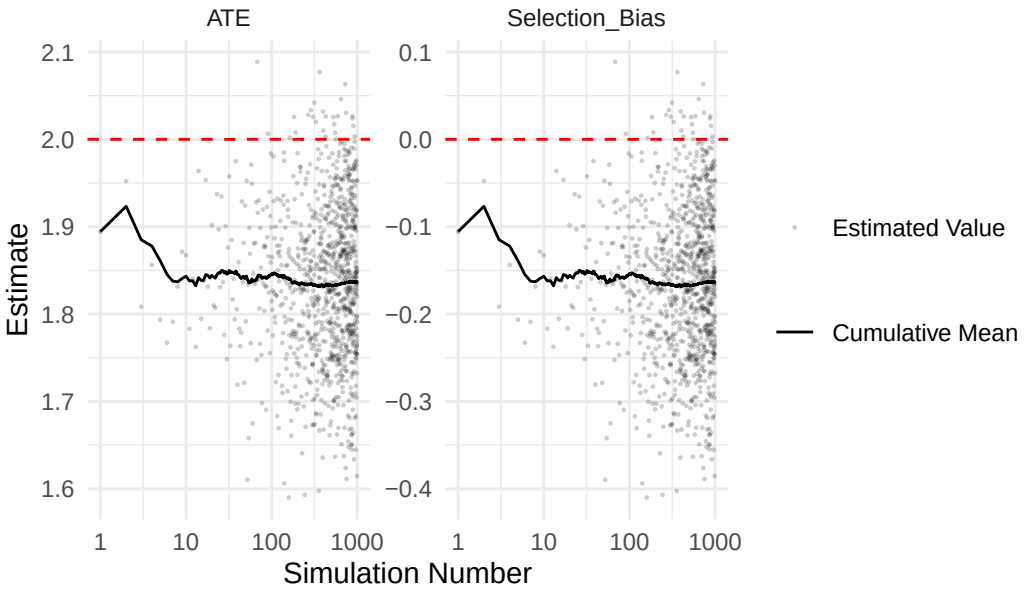
```r
# Prepare data for plotting
results_long <- results |>
    pivot_longer(
        cols = !iter,
        names_to = "Metric",
        values_to = "Value"
    )

# Histogram of ATEs and Selection Biases
ggplot(results_long, aes(x = Value, fill = Metric)) +
    geom_histogram(alpha = 0.5) +
    facet_wrap(~Metric, scales = "free") +
    labs(
        title = "Distribution of Estimated ATEs and Selection Biases",
        x = "Value",
        y = "Count"
    ) +
    theme_minimal()
```



```r
# Cumulative Mean Plot that shows convergence (or lack thereof)
results_long_cumulative <- results_long |>
    group_by(Metric) |>
    mutate(
        cum_mean = cummean(Value)
    )
```

Another way to visualise the convergence to the wrong ATE and a non-zero selection bias is to plot the cumulative mean of the estimates across simulations. This plot shows that as the number of simulations increases, the cumulative mean of the estimated ATEs converges to a value that is different from the true ATE of 2, and the cumulative mean of the selection biases converges to a value that is different from 0.

```r
ggplot(results_long_cumulative, aes(x = iter, y = cum_mean)) +
    geom_hline(
        aes(
            yintercept = ifelse(Metric == "ATE", 2, 0)
        ),
        color = "red",
        linetype = "dashed"
    ) +
    geom_line(aes(linetype = "Cumulative Mean")) +
    geom_point(aes(y = Value, shape = "Estimated Value"),
        alpha = .2,
        size = .5
    ) +
    scale_x_log10() +
    facet_wrap(~Metric, scales = "free_y") +
    labs(
        title = "Cumulative Mean of Estimated ATEs and Selection Biases",
        x = "Simulation Number",
        y = "Estimate"
    ) +
    theme_minimal() +
    theme(legend.title = element_blank())
```



3 Replication

In this section, you will be working with replication data from the study "How to Elect More Women: Gender and Candidate Success in a Field Experiment". The authors designed a field experiment to test whether messages from party leaders can affect women's electoral success in intra party elections. The general belief is that there are two factors that explain why few women are elected. The first factor is related to the so-called 'supply side', with fewer female candidates vying for office. However, particularly among conservatives, voters' biases, also called the 'demand side', may play an important role. In the experiment messages are sent to leaders of precinct-level caucus meetings to see if tackling the demand side, the supply side, or both jointly, can increase the number of women who are elected.

The messages were divided into 4 categories: 1) Placebo control, 2) Supply messages, 3) Demand messages, and 4) Supply+Demand messages. In the data, group 1 is the control group receiving a placebo message unrelated to the aforementioned factors. Groups 2 to 4 represent the different treatment groups with messages relating to both factors.

3.1 Read into R the replication data file `how_to_elect_more_women.dta`. Using `prop_sd_fem2014`, the proportion of state delegates elected from the precinct in 2014 who were women, create a new dummy variable called `sd_onefem2014` that takes a value of 1 if at least one woman was elected within the precinct in 2014, and 0 otherwise. This will be our outcome variable.

Solution

```
data <- read_dta(here("data", "how_to_elect_more_women.dta"))
data$sd_onefem2014 <- ifelse(data$prop_sd_fem2014 > 0, 1, 0)
```

3.2 Using the treatment indicator `condition`, create a balance table that compares the means of the pre-treatment variables `age2012` and `attendees2012` across the four treatment groups. Do you observe any significant imbalances? How could they be addressed in the analysis?

Solution

There are several ways to create a balance table in R. You could run a series of t-tests using `t.test()`, run linear regressions with the pre-treatment variables as outcomes and the treatment indicator as (sole) predictor, or use a package like `modelsummary` to create a more formal balance table. These approaches all have their pros and cons. The `modelsummary` approach is quick and easy, but it doesn't provide inferential statistics when there are more than two treatment groups. The regression and t-test approaches provide inferential statistics, but require more coding. Below are examples of all three approaches. Note that regression will return the mean in the reference group (usually, this should be the control group) as the intercept, and the difference in means for the other treatment groups as coefficients.

```r
# using modelsummary
# subset the data to only include relevant variables
balance_data <- data |>
    select(condition, age2012, attendees2012) |>
    mutate(condition = as_factor(condition))

# when using the modelsummary package, we can create a balance table like this
balance_table_ms <- modelsummary::datasummary_balance(
    # if we want to get balance across all variables in the dataset,
    # we only need to specify a one-sided formula with the treatment
    # indicator on the right
    ~condition,
    # and the dataset
    balance_data,
    output = "markdown",
)

balance_table_ms
```

	Control (N=541)		Supply (N=539)		Demand (N=538)		Supply+Demand (N=538)	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
age2012	55.0	7.0	55.4	6.3	54.9	7.1	54.9	7.2
attendees2012	32.1	33.3	58.6	39.9	57.1	39.3	54.6	38.2

```r
# when using regression, it can be helpful to use a loop or one of the
# apply functionsto avoid having to repeat code multiple times.
balance_table_lm <- lapply(c("age2012", "attendees2012"), function(var) {
    formula <- as.formula(paste(var, "~ condition"))
    estimatr::lm_robust(formula, data = balance_data) |>
        # the broom package makes it easy to tidy model outputs
        # (it turns them into dataframes)
        broom::tidy(conf.int = T)
}) |>
    bind_rows() |>
    select(term, outcome, estimate, std.error, p.value)

knitr::kable(balance_table_lm,
    digits = 3,
    caption = "Balance Table using Regression Approach"
)
```

Table 6: Balance Table using Regression Approach

term	outcome	estimate	std.error	p.value
(Intercept)	age2012	55.034	0.321	0.000
conditionSupply	age2012	0.370	0.432	0.392
conditionDemand	age2012	-0.085	0.458	0.853
conditionSupply+Demand	age2012	-0.161	0.461	0.726
(Intercept)	attendees2012	52.066	1.535	0.000
conditionSupply	attendees2012	6.559	2.388	0.006
conditionDemand	attendees2012	5.000	2.372	0.035
conditionSupply+Demand	attendees2012	2.536	2.334	0.277

```r
# using t-tests
balance_table_ttest <- lapply(c("age2012", "attendees2012"), function(var) {
    formula <- as.formula(paste(var, "~ condition"))

    control_vs_supply <- t.test(formula,
        data = balance_data[balance_data$condition %in% c("Control", "Supply"), ]
    ) |>
        broom::tidy(conf.int = T)

    control_vs_demand <- t.test(formula,
        data = balance_data[balance_data$condition %in% c("Control", "Demand"), ]
    ) |>
        broom::tidy(conf.int = T)

    control_vs_supply_and_demand <- t.test(formula,
        data = balance_data[balance_data$condition %in% c("Control", "Supply+Demand"), ]
    ) |>
        broom::tidy(conf.int = T)

    list(
        control_vs_supply = control_vs_supply,
        control_vs_demand = control_vs_demand,
        control_vs_supply_and_demand = control_vs_supply_and_demand
    ) |>
        bind_rows(.id = "condition") |>
        mutate(Variable = var)
}) |>
    bind_rows() |>
    select(Variable, condition, estimate, p.value)

# you can then create a table from any of these outputs using
# the `kable` package
knitr::kable(balance_table_ttest,
    digits = 3,
    caption = "Balance Table using T-Tests Approach"
)
```

Table 7: Balance Table using T-Tests Approach

Variable	condition	estimate	p.value
age2012	control_vs_supply	-0.370	0.392

age2012	control_vs_demand	0.085	0.853
age2012	control_vs_supply_and_demand	0.161	0.726
attendees2012	control_vs_supply	-6.559	0.006
attendees2012	control_vs_demand	-5.000	0.035
attendees2012	control_vs_supply_and_demand	-2.536	0.278

The `attendees2012` variable shows some significant imbalances between the placebo control group and both the demand and the supply condition. If these imbalances are due to chance, you could control for the variable when estimating treatment effects. However, if these imbalances are systematic (e.g. assignment wasn't random), controlling for them won't solve the problem: the potential outcomes will most certainly also differ systematically between treatment groups, leading to biased estimates of the ATE.

3.3 Using `lm_robust()` from the `estimatr` package, estimate the Average Treatment Effect (ATE) of each treatment group compared to the control group on the outcome variable `sd_onefem2014`. Present your results in a table and interpret them.

Solution

In the placebo group, about 37.5 percent of precincts elected at least one woman. The supply message increased that share by about 5.9 percentage points (%p), the demand message by about 5.7%p and the combined supply and demand message by about 7.8%p. However, only the treatment effect of the combined message is statistically significant at the 5% level.

```r
ate_estimatr <- estimatr::lm_robust(
    sd_onefem2014 ~ as_factor(condition),
    # By default, lm_robust uses HC2 heteroskedasticity-robust standard errors
    se_type = "HC2",
    data = data
)

# create a table using modelsummary
modelsummary::modelsummary(
    list(
        "Any Woman Elected" = ate_estimatr
    ),
    coef_rename = c(
        "(Intercept)" = "Placebo Control Group Mean",
        "as_factor(condition)Supply" = "Supply",
        "as_factor(condition)Demand" = "Demand",
        "as_factor(condition)Supply+Demand" = "Supply+Demand"
    ),
    gof_map = c("nobs", "r.squared"),
    notes = "Heteroskedasticity-robust standard errors in parentheses.",
    stars = TRUE,
    title = "Intent-to-Treat Estimates from Linear Regression Models",
    output = "markdown"
)
```

Table 8: Intent-to-Treat Estimates from Linear Regression Models

	Any Woman Elected
Placebo Control Group Mean	0.375***
	(0.023)
Supply	0.059+
	(0.032)
Demand	0.057+
	(0.033)
Supply+Demand	0.078*
	(0.033)
Num.Obs.	1812
R2	0.003

- + $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

3.4 BONUS: When using a categorical independent variable (like a treatment indicator), classical standard errors can be biased downwards due to heteroscedasticity. Above, we estimated HC2 heteroscedasticity robust standard errors to account for that. An alternative way to estimate inferential statistics without relying on asymptotic assumptions is randomisation inference. On a subset of the data that only includes the placebo and supply+demand conditions, test the sharp null hypothesis of no effect using randomisation inference with 10,000 permutations. Report the p-value and create a histogram of the simulated ATEs. You can assume that the treatment was assigned using complete random assignment.

Hint:

1. Estimate the observed ATE in that subset.
2. Create a loop that repeats the following steps 10,000 times:
    - Shuffle the treatment assignment vector.
    - Estimate the ATE under the permuted treatment assignment and store it.
3. Calculate the two-tailed p-value as the share of permuted ATEs that are as extreme or more extreme than the observed ATE.

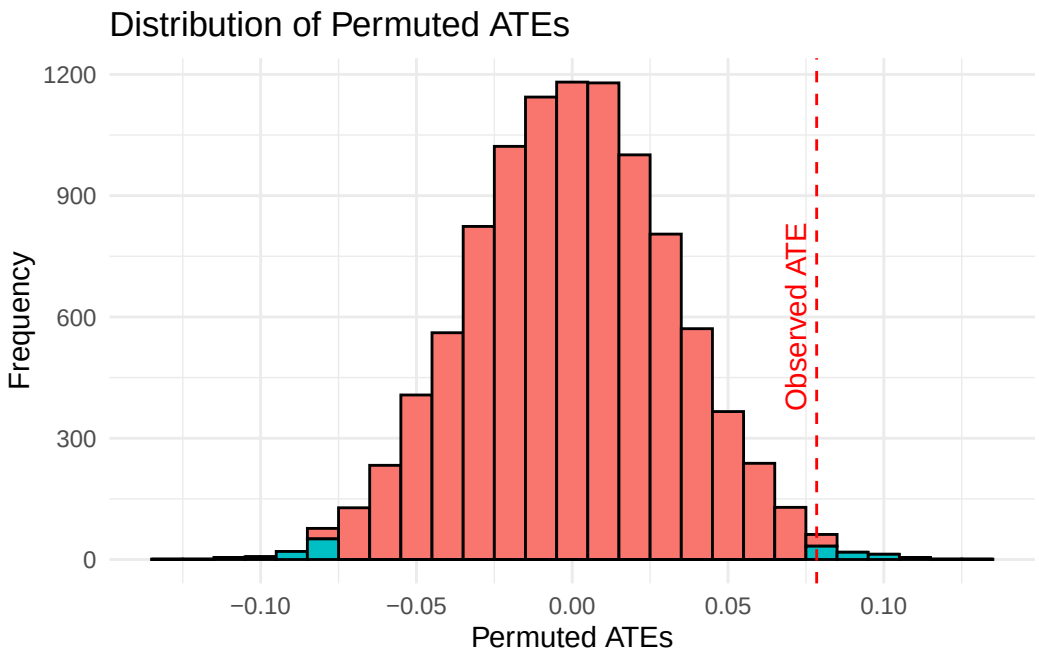
Solution

There are several packages that can be used to conduct randomisation inference in R, such as `ri`, `ri2` or `ritest`. In this solution, to make the mechanics of randomisation inference more transparent, we implement a simple version of randomisation inference from scratch with just 6 lines of code.

```r
# subset the data to only include placebo and supply+demand conditions
data_ri <- data |>
    filter(as_factor(condition) %in% c("Control", "Supply+Demand")) |>
    mutate(
        D = ifelse(as_factor(condition) == "Supply+Demand", 1, 0)
    )

set.seed(2026)
n_permutations <- 10000
observed_ate <- lm(sd_onefem2014 ~ D, data = data_ri)$coefficients["D"]

permuted_ates <- replicate(n_permutations, {
    # shuffle the treatment assignment vector
    permuted_D <- sample(data_ri$D)
    # estimate the ATE under the permuted treatment assignment and store it
    lm(data_ri$sd_onefem2014 ~ permuted_D)$coefficients["permuted_D"]
})

# calculate the two-tailed p-value as the share of permuted ATEs that
# are as extreme or more extreme than the observed ATE
p_value <- mean(abs(permuted_ates) >= abs(observed_ate))
p_value
```

```
[1] 0.0156
```

```r
# plot the distribution of permuted ATEs
ggplot(data.frame(permuted_ates), aes(x = permuted_ates)) +
    geom_histogram(
        aes(
            fill = abs(permuted_ates) >= abs(observed_ate)
        ),
        binwidth = 0.01,
        color = "black"
    ) +
    geom_vline(
        xintercept = observed_ate,
        color = "red",
        linetype = "dashed"
    ) +
    annotate("text",
        x = observed_ate,
        y = 600,
        label = "Observed ATE",
        color = "red",
        angle = 90,
        vjust = -0.5
    ) +
    labs(
        title = "Distribution of Permuted ATEs",
        x = "Permuted ATEs",
        y = "Frequency"
    ) +
    theme_minimal() +
    theme(legend.position = "none")
```



The p-value obtained from randomisation inference is 0.0156, indicating that we can reject the sharp null hypothesis of no effect at $\alpha = 0.05$. This value means that only about 1.56 percent of the ATEs obtained under random permutations of the treatment assignment were as extreme or more extreme than the observed ATE. Note that you may get slightly differnent p-values due to random variation in the permutations.

Feedback

Feedback to the problem set will be provided here.