Problem Set 1 — POLS601

Zhengyu Xiao

2025-10-05

Simulation

```r
set.seed(123)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

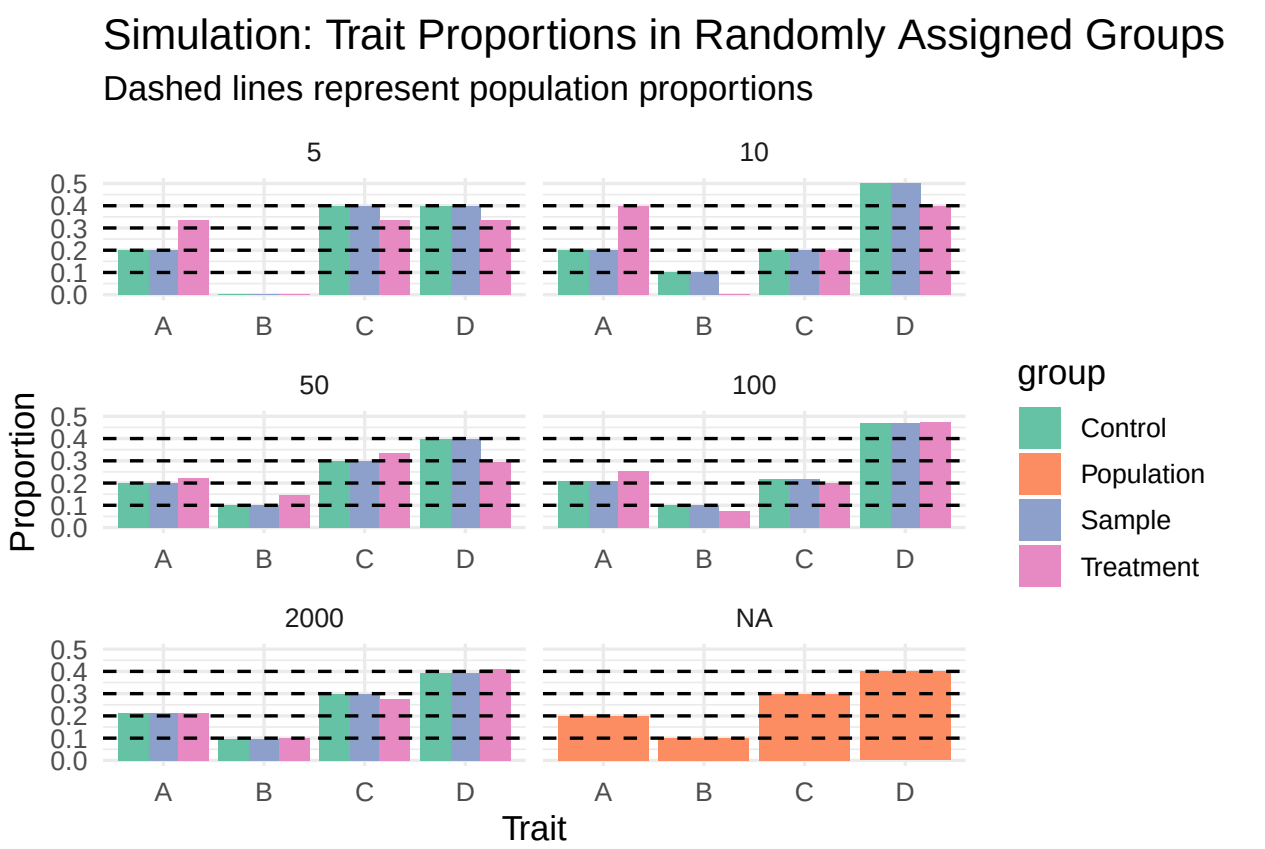
```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v ggplot2   4.0.0     v stringr   1.5.2
## v lubridate 1.9.4     v tibble    3.3.0
## v purrr     1.1.0     v tidyr     1.3.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(ggplot2)

# Randomly samples n observations from a population with some distribution of traits
N <- 100000
traits <- c("A","B","C","D")
p_true  <- c(0.20, 0.10, 0.30, 0.40)

# Create population with traits following true probabilities
```

```r
population <- data.frame(
  id <- 1:N,
  trait <- sample(traits, size = N, replace = TRUE, prob = p_true)
)

# Function to run n times of simulation
simulation <- function(n){
  # sample n observations from population
  sample_data <- population %>% sample_n(n)
  # randomly assign treatment with equal probability
  sample_data$treatment <- rbinom(n, size = 1, prob = 0.5)

  #compute trait proportions for entire sample
  sample_data$trait <- factor(sample_data$trait, levels = traits)
  prop_all <- prop.table(table(sample_data$trait))
  prop_treatment <- prop.table(table(sample_data$trait[sample_data$treatment == 1]))
  prop_control <- prop.table(table(sample_data$trait[sample_data$treatment == 0]))

  #Tidy Data
  data.frame(
    group = rep(c("Sample", "Treatment", "Control"), each = length(traits)),
    trait = rep(traits, 3),
    proportion = c(prop_all, prop_treatment, prop_all),
    n=n
  )
}

# Run simulation for increasing sample sizes
sample_sizes <- c(5, 10, 50, 100, 2000)
result_list <- lapply(sample_sizes, simulation)
sim_results <- rbind(result_list[[1]], result_list[[2]], result_list[[3]], result_list[[4]], result_lis

# Add population proportions for comparison
pop_df <- data.frame(
  group = "Population",
  trait = traits,
  proportion = p_true,
  n = NA
)

# Combine results
plot_data <- rbind(sim_results, pop_df)

# Plot
ggplot(plot_data, aes(x = trait, y = proportion, fill = group)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_wrap(~ n, ncol = 2, scales = "free_x") +
  geom_hline(yintercept = p_true, color = "black", linetype = "dashed") +
  labs(
    title = "Simulation: Trait Proportions in Randomly Assigned Groups",
    subtitle = "Dashed lines represent population proportions",
    x = "Trait",
    y = "Proportion"
```

```
  ) +
  theme_minimal(base_size = 13) +
  scale_fill_brewer(palette = "Set2")
```




```
# So from the simulation it is clear that as n increases, the distribution of traits in the sample, tre
```

Data Analysis

```
voting <- read.csv("voting.csv")

class(voting$message)
```

```
## [1] "character"
```

```
class(voting$voted)
```

```
## [1] "integer"
```

```
class(voting$birth)
```

```
## [1] "integer"
```

```r
# Treatment is the social pressure message, it is a discrete variable, the data type is character/strin

# Create a new treatment variable in your data frame that is a binary version of the existing treatment
voting$treatment <- ifelse (voting$message == "yes", 1, 0)

# Compute the average outcome for the treatment group and the average outcome for the control group. In

avg_outcome <- voting %>%
  group_by (treatment) %>%
  summarise (avg_outcome = mean (voted, na.rm = TRUE))

avg_outcome_treatment <- avg_outcome$avg_outcome[avg_outcome$treatment == 1]
avg_outcome_control <- avg_outcome$avg_outcome[avg_outcome$treatment == 0]

avg_outcome_treatment
```

```
## [1] 0.3779482
```

```r
avg_outcome_control
```

```
## [1] 0.2966383
```

```r
# Interpret: The average turnout in the treatment group, who received the social pressure message, was

# Use brackets to subset the data frame and create two new data frames, one for the treatment group and
treatment_data <- voting[voting$treatment == 1, ]
control_data <- voting[voting$treatment == 0, ]

# What is the average birth year for the treatment and control groups?
avg_birth_treatment <- mean(treatment_data$birth)
avg_birth_control <- mean(control_data$birth)
avg_birth_treatment
```

```
## [1] 1956.147
```

```r
avg_birth_control
```

```
## [1] 1956.186
```

```r
# What is the estimated average causal effect for this experiment? Provide the calculated average effec
estimated_effect <- avg_outcome_treatment - avg_outcome_control
estimated_effect
```

```
## [1] 0.08130991
```

```r
# Interpret: exposure to the social pressure mailing substantially increased the likelihood of voting,

# Suppose we wanted to claim that the estimated causal effect is an estimated effect for the entire U.S

# To treat the estimated causal effect from this experiment as the causal effect for the entire U.S. vo
```