

Introduction

The goal of this exercise is to get a feel for running and designing real experiments and how important the framing of a question can be to the outcome of the study. We will design and conduct our own versions of a well-known psychology experiment that investigates exactly this. Finally, we implement statistical tests learned in class to get a better understanding of how they work in the real world, and considerations towards statistical power.

The Experiment

For this lab, we will be working with the Tversky and Kahneman (1981)¹ paper that studied the effects of framing on participant response and risk aversion. Read through the paper, understanding the effect that they are trying to elicit and the setting.

1. Identify and list the four components of the experimental design they used, i.e.

- The Nature of the Treatments.
- Choice of the Experimental Units.
- Manner of Assigning Units to Treatments
- The Nature of the Response

2. Think of another question that elicits this same “framing” effect. Use a similar approach, but change the setting. Keep in mind how language and specific words can influence this. Do you think your question would elicit a smaller or larger effect size?

¹Tversky, A., & Kahneman, D. (1981). *The framing of decisions and the psychology of choice*. *Science*, 211(4481), 453-458.

Conducting Your Own Experiment

Use your new question to conduct your own framing experiment in class. Go around asking your two versions of the question to all others present, and collect data on their responses. Keep the following questions in mind, and write very brief answers explaining your choice in each.

3. How are you going to ask the question? Consider options such as Google Forms, e-mails, writing on paper slips, or verbally asking. How do you think the mode of the question affects the response?
4. Are there other factors that may be relevant? Including a time-limit on answering? Using percentages instead of numbers? Adding a default such as “Most people choose A”? Including a “How confident are you in your response” score? Consider other ways of capturing more information or eliciting the effect.
5. How are you planning to assign units to treatments? Tossing a coin? Using a random number generator? Do you experiment on pairs instead, one of each being in the treatment group and the other control?

Data Analysis

We will analyze both the original dataset as well as our own collected data. Ensure that you tabulate your data into a `.csv` file. This can be done by entering your data into Google Sheets or Microsoft Excel before saving as `.csv`.

We will be using data collected from the Many Labs study (2020), which implemented the same question as Tversky and Kahneman. The experiment was the same in most regards, and they expanded their sample to 6,344 participants recruited from 36 different sources including university subject pools, Amazon Mechanical Turk, Project Implicit, and other sources.

The dataset is stored in a `.csv` file. As earlier, use the `readr` package, which is included inside the `tidyverse`. If you haven't installed the `tidyverse` before, you can do so by running `install.packages("tidyverse")` once.

```
# load tidyverse (includes readr for CSVs)
library(tidyverse)
```

```
# load the data into R
framing <- read_csv("https://stat158.berkeley.edu/spring-2026/data/framing/framing.csv")
```

```
# load your collected data into R as well
```

```
#my_data <-
```

Statistical tests

Non-parametric Methods: Randomization Tests

6. We shall first implement this for the larger ManyLabs dataset provided. In order to conduct a randomization test, we will be working under the Potential Outcomes framework.

Q: What assumptions are required by the permutation test? What is random in this setup?

A:

Modify the dataset to reflect two columns, `Y_1` and `Y_0`, representing the potential outcomes. Note that this will have missing values.

```
# convert data to potential outcomes
```

As in class, we will be testing the null hypothesis that (some people call it the **Fisher Sharp Null Hypothesis**)

$$Y_i(1) - Y_i(0) = 0 \text{ for any } i.$$

Under the **Fisher Sharp Null Hypothesis**, impute the missing potential outcomes.

```
# Fill in the missing values
```

We now implement the randomization test from class. The following function evaluates the distribution of the test statistic under the null by repeatedly randomizing who receives treatment.

```

# Randomization distribution

rand_stats <- function(schedule, d_i, stat = "diff in means", reps = 1000) {

  # replicate the schedule reps times and stack them on top of one another
  randomized_exp_df <- map_dfr(1:reps, ~ schedule, .id = "experiment") |>
    mutate(experiment = factor(experiment, levels = as.character(1:reps), ordered
= TRUE),
          d_i = c(replicate(n = reps, sample(d_i))), # create random assignments
          y_i = Y_1 * d_i + Y_0 * (1 - d_i)) |> # find observed responses
    arrange(experiment)

  # calculate test statistic for every random assignment
  if (stat == "diff in means") {
    stats <- randomized_exp_df |>
      group_by(experiment, d_i) |>
        summarize(ybar = mean(y_i), # average within each group within each experiment
                  .groups = "drop_last") |>
        summarize(ATE_hat = diff(ybar),
                  .groups = "drop") |> # take the difference between the two groups mean
    pull()
  } else {
    stop("Statistic not implemented")
  }

  return(stats)
}

```

7. Fill in your own version of `schedule` and `d_i` based on the dataset we are working with. Proceed to then calculate a p-value. See the [Stat 158 testing code page](#).

```

# Randomization Test

# Fill these in
# my_null_schedule <-
#
# my_d_i <-
#
# my_ATE_obs <-
#
# calculate the test statistic distribution
stats <- rand_stats(schedule = my_null_sched, d_i = my_d_i)

```

```
# calculate p-value
mean(abs(stats) > my_ATE_obs)
```

8. Repeat these steps for your own collected dataset. Convert it to a filled-in potential outcomes style table, generate the randomization distribution and calculate the p-value.

```
# Repeat on your own data
```

Model-based Inference: The Z-Test

9. Before we conduct our z-test, we should be explicit about our hypothesis and assumptions. Answer the following questions in brief, but complete sentences.

Q: What is the null hypothesis of our z-test in the context of our specific problem? Is it different from that of the permutation test? **A:**

Q: What assumptions are required by the z-test?

A:

Note that though we are working with binary data, the proportion of 1's can often be well approximated by a normal distribution, particularly with larger sample sizes. Thus a z-test is still valid in such settings.

10. Fortunately, R has a built-in function, called `prop.test`, that conducts a two-sample z-test on such binary data. Run the test on the Many Labs data in the chunk below and report the p-value. Be sure to add `correct = FALSE` in the arguments.

```
# conduct a two-sample z-test
# ex: prop.test((p1, p2), (n1, n2), correct = FALSE)
```

Run this on your own collected data as well.

```
# conduct a two-sample z-test
```

Q: What can you conclude from the above two test results?

A:

Q: What do you think this tells you about the power of the z-test and the normal approximation?

A:

