Problem set 1

2025-09-26

```
library(tidyverse)
```

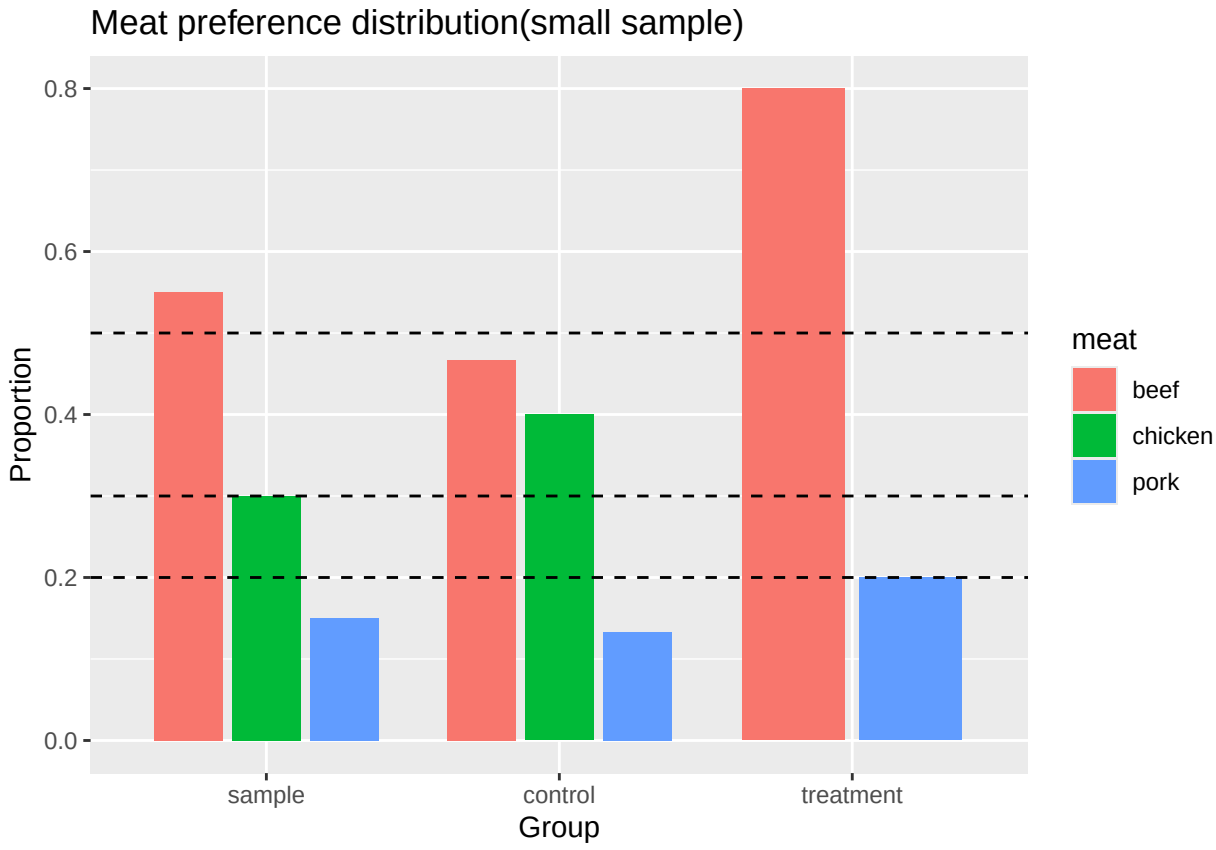
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ 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
```

```
library(dplyr)
```

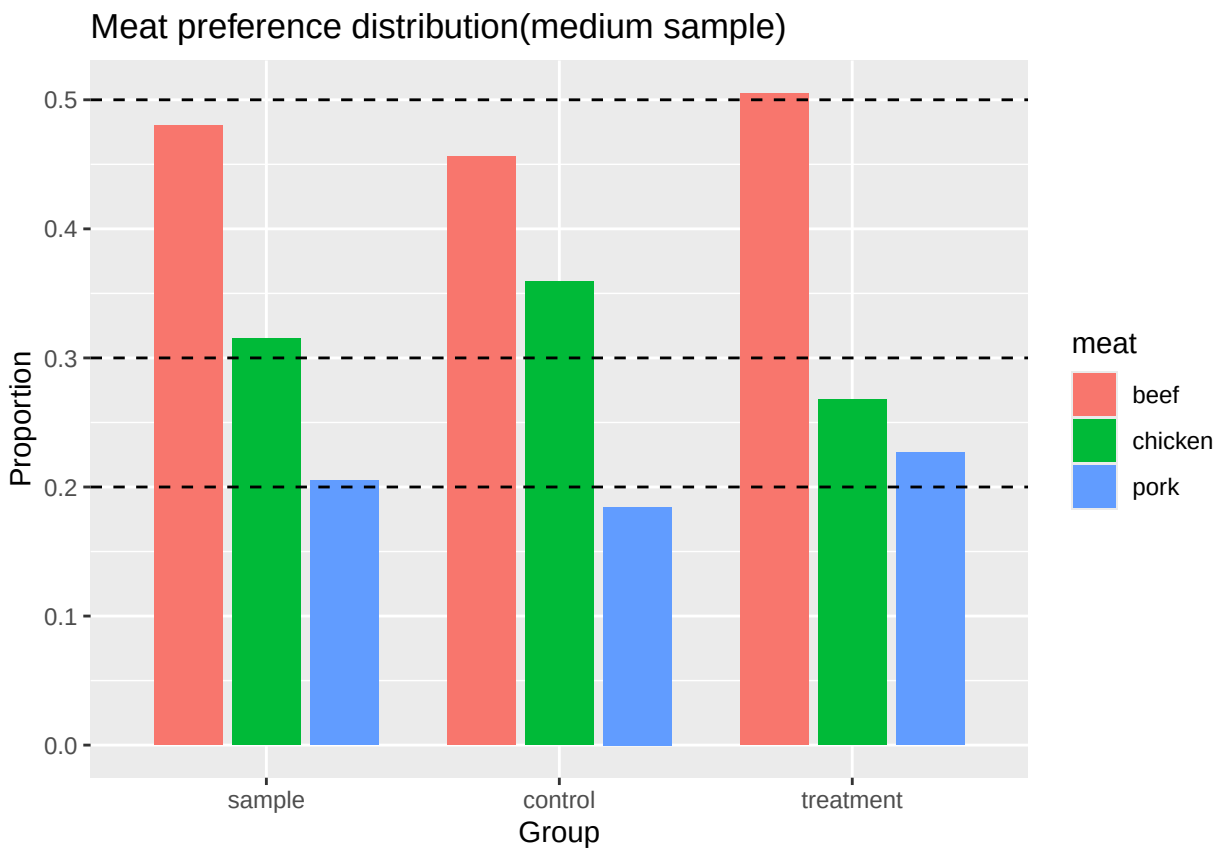
Simulation

```
# Create the population of 1000 individuals.
# Var. 'meat' represents each individual's favorite kind of meat,
# drawn with probabilities: beef 50%, pork 20%, chicken 30%.
population <- tibble(
                  meat = sample(
                    c("beef", "pork", "chicken"),
                    1000, # population size
                    replace = TRUE,
                     prob = c(0.5, 0.2, 0.3)),# distribution of preferences
                  id = 1:1000)

#set up a simulation
# n = sample size per iteration
# n_iter = times of iterations = 1000
simulation <- function(n) {



# Draw a random sample of size n from the population
# Randomly assign sampled individuals into treatment/control groups
sample_df <- population %>%
  slice_sample(n = n) %>%
  mutate(group = sample(c("treatment","control"),size = n,replace = TRUE))

#Count the preference portion in whole sample
  p_sample <- sample_df %>%
  mutate(group = "sample") %>%
```

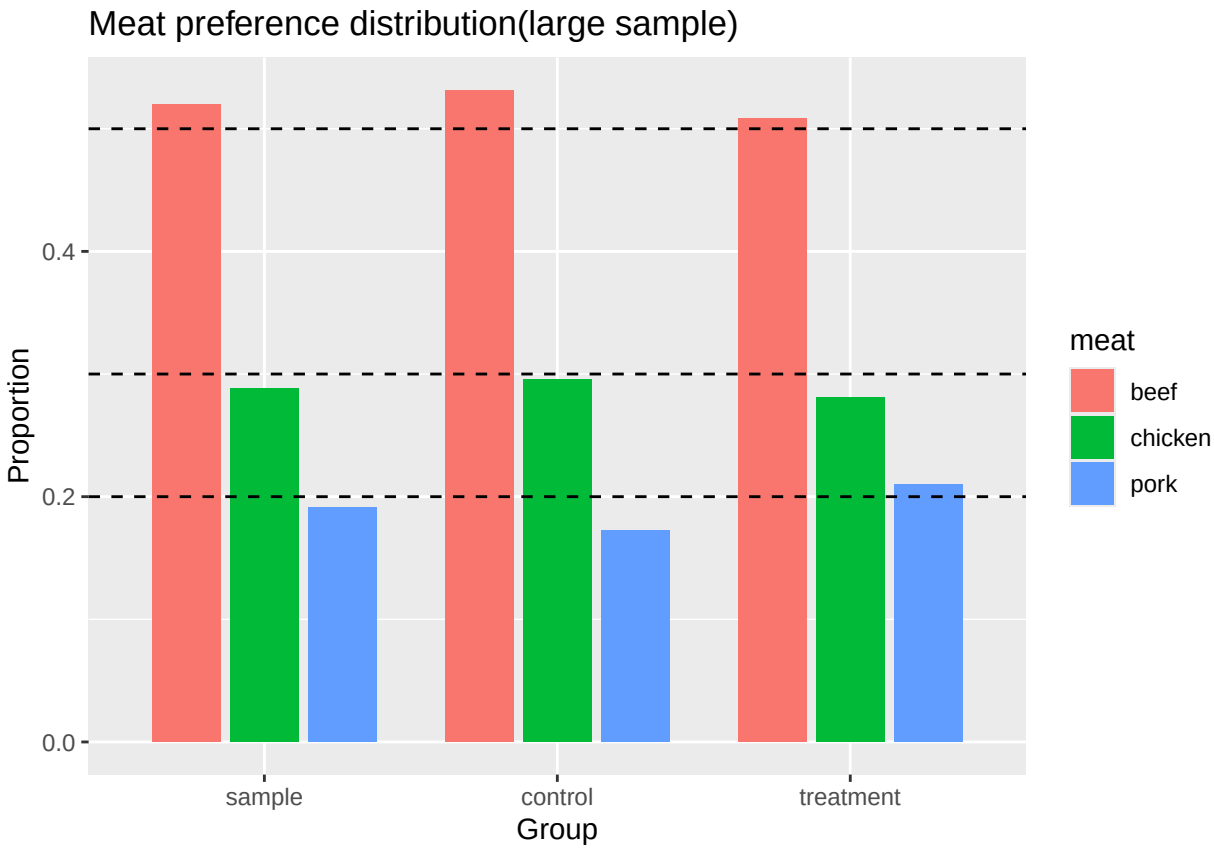
```r
  count(group, meat, name = "count_in_sample") %>%
  mutate(prop = count_in_sample / sum(count_in_sample))

#Count the preference portion in each group
  p_groups <- sample_df %>%
      group_by(group) %>%
      count(meat, name = "count_in_group") %>%
      mutate(prop = count_in_group / sum(count_in_group)) %>%
      ungroup()

    p_final <- bind_rows(p_sample, p_groups) %>%
      mutate(
        group = factor(group, levels = c("sample", "control", "treatment")))


    return(p_final)
}

#run the simulation with a small sample
sample_small <- simulation(20)

ggplot(sample_small, aes(x = group, y = prop, fill = meat)) +
  geom_col(position = position_dodge(width = 0.8), width = 0.7) +
  geom_hline(yintercept = c(0.5, 0.2, 0.3), linetype = "dashed") +
  labs(y = "Proportion", x = "Group",
       title = "Meat preference distribution(small sample)")
```



```r
#run the simulation with a medium sized sample
sample_mid <- simulation(200)

ggplot(sample_mid, aes(x = group, y = prop, fill = meat)) +
  geom_col(position = position_dodge(width = 0.8), width = 0.7) +
  geom_hline(yintercept = c(0.5, 0.2, 0.3), linetype = "dashed") +
  labs(y = "Proportion", x = "Group",
       title = "Meat preference distribution(medium sample)")
```




```r
#run the simulation with a large sized sample
sample_large <- simulation(700)

ggplot(sample_large, aes(x = group, y = prop, fill = meat)) +
  geom_col(position = position_dodge(width = 0.8), width = 0.7) +
  geom_hline(yintercept = c(0.5, 0.2, 0.3), linetype = "dashed") +
  labs(y = "Proportion", x = "Group",
       title = "Meat preference distribution(large sample)")
```



Data Analysis

1.

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

```
## 'data.frame':    229444 obs. of  3 variables:
##  $ birth  : int  1981 1959 1956 1939 1968 1967 1941 1969 1967 1961 ...
##  $ message: chr  "no" "no" "no" "yes" ...
##  $ voted  : int  0 1 1 1 0 0 1 0 1 1 ...
```

The treatment variable is *message*. It is discrete and its data type is *character*.

2.

```
#create a binary version of message
voting <- voting %>%
  mutate(message_bi = ifelse(message == "yes" ,1,0))
table(voting$message_bi)
```

```
##
##      0      1
## 191243  38201
```

3.

```
voting_summary <- voting %>%
  group_by(message_bi) %>%
  summarise(avg_voted = mean(voted))
print(voting_summary)
```

```
## # A tibble: 2 x 2
##   message_bi avg_voted
##        <dbl>     <dbl>
## 1          0     0.297
## 2          1     0.378
```

The result shows that about 29.7% of the voters in the control group (did not receive social pressure message) voted in 2006 election, while about 37.8% of the voters in the treatment group (received social pressure message) voted in 2006 election.

4.

```
#create subsets of voting dataset by control/treatment
voting_control <- voting[voting$message_bi == 0,]
voting_treatment <- voting[voting$message_bi == 1,]
```

5.

```
mean(voting_control$birth)
```

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

```
mean(voting_treatment$birth)
```

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

6.

Refer to question 3, the calculated average effect = 0.3779482 - 0.2966383 approximately equals to 8.1%. Which means, receiving a message of social pressure can increase the voter turnout by 8.1%.

7.

If we want to claim that the estimated causal effect is an estimated effect for the entire U.S. population, we need the assumption that the sample in the experiment must be representative of the whole U.S. population.