exercise1

Xiang Li

2024/2/6

2

```r
data = data.frame(id = 1:10, y_chocolate = c(4, 4, 6, 5, 6, 5, 6, 7, 5, 6), y_vanilla = c(1,
    3, 4, 5, 5, 6, 8, 6, 3, 5))
```

2a

```r
data$causal_eff = data$y_chocolate - data$y_vanilla
```

2d

```r
apply(data[, 2:4], 2, FUN = mean)
```

```
## y_chocolate   y_vanilla  causal_eff
##         5.4         4.6         0.8
```

3

3a

```r
data$obs_x = ifelse(data$y_vanilla >= data$y_chocolate, "vanilla", "chocolate")
data$obs_y = pmax(data$y_chocolate, data$y_vanilla)
```

3b

```r
mean(data[data$obs_x == "chocolate", "obs_y"]) - mean(data[data$obs_x == "vanilla",
    "obs_y"])
```

```
## [1] -0.9047619
```

Not same.

3c

Exchangeability assumption and positivity assumption.

3d

Add some people who prefer chocolate but choose vanilla or prefer vanilla but choose chocolate.

4

```
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
```

4a

```
## we are doing something *random* so let's set a seed so we always observe the
## same result each time we run the code
set.seed(11)
data_observed = select(mutate(data, exposure = case_when(rbinom(10, 1, 0.5) == 1 ~
    "chocolate", TRUE ~ "vanilla"), observed_outcome = case_when(exposure == "chocolate" ~
    y_chocolate, exposure == "vanilla" ~ y_vanilla)), id, exposure, observed_outcome)
```

```
mean(data_observed[data_observed$exposure == "chocolate", "observed_outcome"]) -
    mean(data_observed[data_observed$exposure == "vanilla", "observed_outcome"])
```

```
## [1] 0.6190476
```

4b

The results are not same. Because the sample size is too small, we can't get effectively randomized results.

4c

Yes, the assumptions met.

5

5a

```
data = data.frame(id = 1:10, y_spoiledchocolate = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
    y_chocolate = c(4, 4, 6, 5, 6, 5, 6, 7, 5, 6), y_vanilla = c(1, 3, 4, 5, 5, 6,
        8, 6, 3, 5))
data = mutate(data, causal_effect = y_chocolate - y_vanilla)
set.seed(11)
data_observed = mutate(data, exposure_unobserved = case_when(rbinom(10, 1, 0.25) ==
    1 ~ "chocolate (spoiled)", rbinom(10, 1, 0.25) == 1 ~ "chocolate", TRUE ~ "vanilla"),
    observed_outcome = case_when(exposure_unobserved == "chocolate (spoiled)" ~ y_spoiledchocolate,
        exposure_unobserved == "chocolate" ~ y_chocolate, exposure_unobserved ==
            "vanilla" ~ y_vanilla), exposure = case_when(exposure_unobserved %in%
        c("chocolate (spoiled)", "chocolate") ~ "chocolate", exposure_unobserved ==
        "vanilla" ~ "vanilla"))
data_observed = select(data_observed, id, exposure, observed_outcome)
```

```
mean(data_observed[data_observed$exposure == "chocolate", "observed_outcome"]) -
    mean(data_observed[data_observed$exposure == "vanilla", "observed_outcome"])
```

```
## [1] -1.916667
```

5b

Consistency assumption (Well defined exposure).

6

6a

```
data = data.frame(id = 1:10, partner_id = c(1, 1, 2, 2, 3, 3, 4, 4, 5, 5), y_chocolate_chocolate = c(4,
    4, 6, 5, 6, 5, 6, 7, 5, 6), y_chocolate_vanilla = c(6, 6, 8, 7, 8, 7, 8, 9, 7,
    8), y_vanilla_chocolate = c(3, 5, 6, 7, 7, 8, 10, 8, 5, 7), y_vanilla_vanilla = c(1,
    3, 4, 5, 5, 6, 8, 6, 3, 5))
set.seed(11)
data_observed = mutate(data, exposure = case_when(rbinom(10, 1, 0.5) == 1 ~ "chocolate",
    TRUE ~ "vanilla"), exposure_partner = c("vanilla", "vanilla", "vanilla", "chocolate",
    "chocolate", "vanilla", "vanilla", "vanilla", "vanilla", "chocolate"), observed_outcome = case_when
    "chocolate" & exposure_partner == "chocolate" ~ y_chocolate_chocolate, exposure ==
    "chocolate" & exposure_partner == "vanilla" ~ y_chocolate_vanilla, exposure ==
    "vanilla" & exposure_partner == "chocolate" ~ y_vanilla_chocolate, exposure ==
    "vanilla" & exposure_partner == "vanilla" ~ y_vanilla_vanilla))
data_observed = select(data_observed, id, exposure, observed_outcome)
```

```
mean(data_observed[data_observed$exposure == "chocolate", "observed_outcome"]) -
    mean(data_observed[data_observed$exposure == "vanilla", "observed_outcome"])
```

```
## [1] 1.761905
```

6b

Consistency assumption (Interference).