QTM 220 HW #4

Author

Veronica Vargas

Exercise #1

Fire VS Water

```
library(tidyverse)
```

```
Warning: package 'tidyverse' was built under R version 4.3.3
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✓ dplyr     1.1.3     ✓ readr     2.1.4
✓ forcats   1.0.0     ✓ stringr   1.5.0
✓ ggplot2   3.4.3     ✓ tibble    3.2.1
✓ lubridate 1.9.2     ✓ tidyr     1.3.0
✓ purrr     1.0.2
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(dplyr)

fireVSwater_pokemon <- read.csv("C:/Users/13015/OneDrive - Emory University/Documents/Fall 2024/QTM
         220/fireVSwater_pokemon.csv")
```

```
head(fireVSwater_pokemon)
```

```
  weight_kg  type generation
1      96.0 water          6
2       7.9  fire          2
3      45.5 water          1
4      90.0 water          1
5      39.5 water          3
6       9.9  fire          5
```

```
summary(fireVSwater_pokemon)
```

```
   weight_kg          type            generation
 Min.   :  0.30   Length:127         Min.   :1.000
 1st Qu.: 11.75   Class :character   1st Qu.:2.000
 Median : 28.00   Mode  :character   Median :3.000
 Mean   : 55.01                      Mean   :3.528
 3rd Qu.: 59.00                      3rd Qu.:5.000
 Max.   :398.00                      Max.   :7.000
```

(a) Showing Balanced Groups

```
fireVSwater_pokemon %>%
  group_by(generation, type) %>%
  summarise(
```

```
    count = n(),
    weight = mean(weight_kg, na.rm = T)) %>%
  ungroup()
```

```
`summarise()` has grouped output by 'generation'. You can override using the
`.groups` argument.
```

```
# A tibble: 14 × 4
   generation type  count weight
        <int> <chr> <int>  <dbl>
 1          1 fire      8   50.9
 2          1 water    21   55.4
 3          2 fire      7   72.1
 4          2 water    12   53.1
 5          3 fire      3   41.3
 6          3 water    20   82.7
 7          4 fire      2   38.5
 8          4 water     7   60.4
 9          5 fire     10   56.8
10          5 water    10   39.3
11          6 fire      6   30.4
12          6 water     7   54.8
13          7 fire      3   80.4
14          7 water    11   20.9
```

(b) Summary Statistics w/ Plotting Evolution

```
pokemon_summary <- fireVSwater_pokemon %>%
  group_by(generation, type) %>%
  summarise(
    count = n(),
    mean_weight = mean(weight_kg, na.rm = T),
    sd_weight = sd(weight_kg, na.rm = T)) %>%
  ungroup()
```

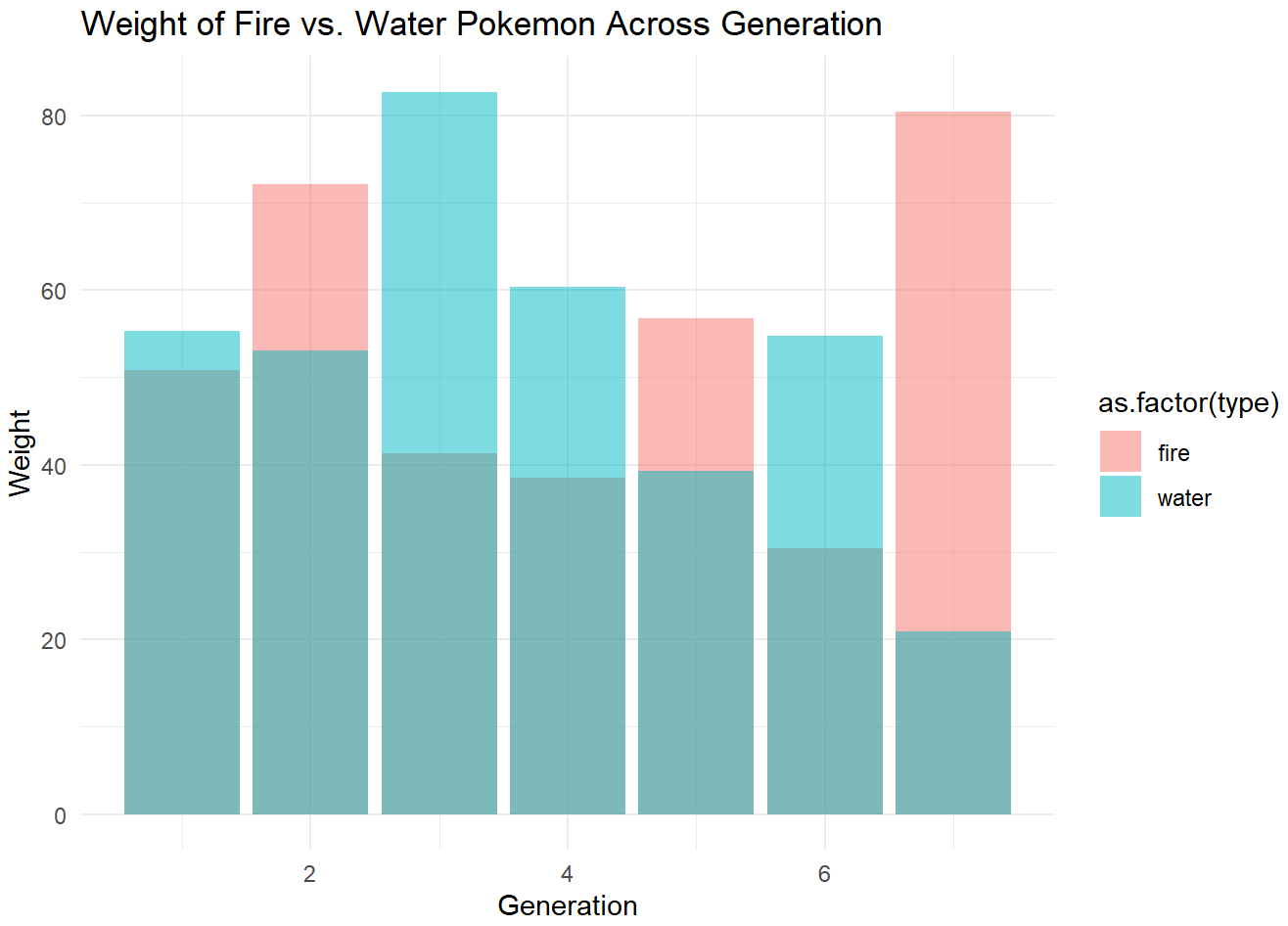
```
`summarise()` has grouped output by 'generation'. You can override using the
`.groups` argument.
```

```
pokemon_summary
```

```
# A tibble: 14 × 5
   generation type  count mean_weight sd_weight
        <int> <chr> <int>       <dbl>     <dbl>
 1          1 fire      8        50.9      49.9
 2          1 water    21        55.4      52.1
 3          2 fire      7        72.1      87.0
 4          2 water    12        53.1      72.9
 5          3 fire      3        41.3      33.9
 6          3 water    20        82.7     111.
 7          4 fire      2        38.5      23.3
 8          4 water     7        60.4     122.
 9          5 fire     10        56.8      98.0
10          5 water    10        39.3      37.9
11          6 fire      6        30.4      27.2
12          6 water     7        54.8      69.1
13          7 fire      3        80.4     114.
14          7 water    11        20.9      23.4
```

```
ggplot(pokemon_summary, aes(x = generation, y = mean_weight,
                            fill = as.factor(type))) +
  geom_bar(stat = "identity", position = "identity", alpha = 0.5) +
  labs(title = "Weight of Fire vs. Water Pokemon Across Generation",
```

```
       x = "Generation",
       y = "Weight") +
  theme_minimal()
```



Here we see that, on average, weight decreases for both water and fire pokemon across generations. That said, there are consistent spikes in weight between generations.

(c) Difference in Mean Weight

```
fire <- fireVSwater_pokemon[fireVSwater_pokemon$type == "fire",]
water <- fireVSwater_pokemon[fireVSwater_pokemon$type == "water",]

mean(fire$weight_kg)
```

```
[1] 53.95128
```

```
mean(water$weight_kg)
```

```
[1] 55.47955
```

```
mean(fire$weight_kg) - mean(water$weight_kg)
```

```
[1] -1.528263
```

This estimator is looking at the difference in mean weight between fire and water pokemon across all generations.

(d) Difference in Mean Weight Across Generations

```
fire <- fireVSwater_pokemon[fireVSwater_pokemon$type == "fire",]
water <- fireVSwater_pokemon[fireVSwater_pokemon$type == "water",]

water <- water %>%
  group_by(generation) %>%
  summarise(avg_weight_water = mean(weight_kg, na.rm = TRUE), n_water = n()) %>%
  ungroup()

fire <- fire %>%
  group_by(generation) %>%
  summarise(avg_weight_fire = mean(weight_kg, na.rm = TRUE), n_fire = n()) %>%
  ungroup()

df <- full_join(water, fire, by = "generation") %>%
  mutate(mean_diff = avg_weight_fire - avg_weight_water)

mean(df$mean_diff)
```

```
[1] 0.5595176
```

This estimator is looking at the average difference in mean weight between fire and water pokemon for each generation.

(e) Difference in Mean Weight Across Generations w/ Fire Pokemon Focus

```
fire <- fireVSwater_pokemon[fireVSwater_pokemon$type == "fire",]
water <- fireVSwater_pokemon[fireVSwater_pokemon$type == "water",]

water <- water %>%
  group_by(generation) %>%
  summarise(avg_weight_water = mean(weight_kg, na.rm = TRUE), n_water = n()) %>%
  ungroup()

fire <- fire %>%
  group_by(generation) %>%
  summarise(avg_weight_fire = mean(weight_kg, na.rm = TRUE), n_fire = n()) %>%
  ungroup()

df <- full_join(water, fire, by = "generation")

(1/39) * sum(df$n_fire * (df$avg_weight_fire - df$avg_weight_water))
```

```
[1] 3.504875
```

This estimator is looking at the weighted difference in means between fire and water pokemon for each generation with a focus on fire pokemon.

(f) Comparing Estimators

None of these estimates are the same. This is because the groups are not balanced and by choosing to weight certain subgroups, our estimates will essentially be different.

(g) Bootstrapping Estimator

```
set.seed(42)

n <- 10000
diff_boot <- rep(NA, n)
```

```
for(i in 1:n){

  sample_boot <- fireVSwater_pokemon[sample(1:nrow(fireVSwater_pokemon), nrow(fireVSwater_pokemon),
         replace = TRUE),]
  fire_boot <- sample_boot[sample_boot$type == "fire",]
  water_boot <- sample_boot[sample_boot$type == "water",]

  water_boot <- water_boot %>%
    group_by(generation) %>%
    summarise(avg_weight_water = mean(weight_kg, na.rm = TRUE), n_water = n()) %>%
    ungroup()

  fire_boot <- fire_boot %>%
    group_by(generation) %>%
    summarise(avg_weight_fire = mean(weight_kg, na.rm = TRUE), n_fire = n()) %>%
    ungroup()

  df <- full_join(water_boot, fire_boot, by = "generation")

  diff_boot[i] <- (1/39) * sum(df$n_fire * (df$avg_weight_fire - df$avg_weight_water), na.rm = TRUE)
}
```

```
lower.bound <- quantile(diff_boot, 0.025)
upper.bound <- quantile(diff_boot, 0.975)

print(paste0("The Bootstrapped 95% CI is {", lower.bound,", ",upper.bound,"}"))
```

```
[1] "The Bootstrapped 95% CI is {-23.2395722804476, 33.2973470954213}"
```

Exercise #4

Voter Turnout Experiment Analysis

```
GGLsample <- read.csv("C:/Users/13015/OneDrive - Emory University/Documents/Fall 2024/QTM
         220/GGLsample.csv")

head(GGLsample)
```

```
       X    sex  yob g2000 g2002 g2004 p2000 p2002 p2004  treatment cluster
1 124262   male 1965   yes   yes   yes   yes    no    No    Control    3581
2  19788   male 1955   yes   yes   yes   yes   yes    No    Control     580
3  92679 female 1980   yes    no   yes    no   yes    No    Control    2670
4 258419 female 1949   yes   yes   yes    no    no    No    Control    7520
5 118408 female 1960   yes    no   yes    no    no    No    Control    3411
6 264080 female 1976   yes    no   yes    no    no    No  Neighbors    7683
  voted  hh_id hh_size numberofnames p2004_mean g2004_mean age bintreat
1   Yes  64453       2            21  0.1904762  0.9047619  41        0
2    No  10439       2            21  0.1428571  1.0000000  51        0
3    No  48056       4            21  0.2380952  0.8571429  26        0
4    No 135357       2            21  0.6666667  1.0000000  57        0
5    No  61389       2            21  0.3333333  0.9047619  46        0
6    No 138278       2            21  0.1904762  0.9523810  30        1
```

```
summary(GGLsample)
```

```
       X              sex                yob          g2000
 Min.   :    65   Length:5000        Min.   :1911   Length:5000
 1st Qu.: 85927   Class :character   1st Qu.:1946   Class :character
 Median :175079   Mode  :character   Median :1956   Mode  :character
```

```
 Mean   :174079                     Mean   :1956
 3rd Qu.:262877                     3rd Qu.:1965
 Max.   :344066                     Max.   :1986
    g2002              g2004              p2000              p2002
 Length:5000        Length:5000        Length:5000        Length:5000
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character



    p2004             treatment            cluster         voted
 Length:5000        Length:5000        Min.   :    3   Length:5000
 Class :character   Class :character   1st Qu.: 2473   Class :character
 Mode  :character   Mode  :character   Median : 5078   Mode  :character
                                       Mean   : 5048
                                       3rd Qu.: 7648
                                       Max.   :10000
     hh_id            hh_size      numberofnames     p2004_mean
 Min.   :    37   Min.   :1.000   Min.   :14.00   Min.   :0.0000
 1st Qu.: 44500   1st Qu.:2.000   1st Qu.:21.00   1st Qu.:0.1847
 Median : 91393   Median :2.000   Median :21.00   Median :0.2857
 Mean   : 90857   Mean   :2.181   Mean   :20.93   Mean   :0.3049
 3rd Qu.:137656   3rd Qu.:2.000   3rd Qu.:21.00   3rd Qu.:0.4286
 Max.   :179989   Max.   :7.000   Max.   :21.00   Max.   :0.9048
   g2004_mean          age            bintreat
 Min.   :0.5238   Min.   :20.00   Min.   :0.0000
 1st Qu.:0.9048   1st Qu.:41.00   1st Qu.:0.0000
 Median :0.9524   Median :50.00   Median :0.0000
 Mean   :0.9227   Mean   :50.25   Mean   :0.1656
 3rd Qu.:1.0000   3rd Qu.:60.00   3rd Qu.:0.0000
 Max.   :1.0000   Max.   :95.00   Max.   :1.0000
```

```
GGLsample <- GGLsample %>%
  mutate(binvote = case_when(
    voted == "Yes" ~ 1,
    voted == "No" ~ 0))
```

(a) CATE(male)

```
df <- GGLsample %>%
  group_by(sex) %>%
  summarise(
    N_Treated = sum(bintreat == 1),
    N_Control = sum(bintreat == 0),
    Mean_Treated = mean(binvote[bintreat == 1]),
    Mean_Control = mean(binvote[bintreat == 0]),
    CATE = Mean_Treated - Mean_Control
  ) %>%
  ungroup()

df$CATE[df$sex == "male"]
```

```
[1] 0.06221803
```

This estimator is not causally identified because the treatment is not conditioning on the sex.

(b) Bootstrapped CATE(male)

```
set.seed(42)

n <- 10000
```

```
diff_boot <- rep(NA, n)

for(i in 1:n){
  sample_boot <-  GGLsample[sample(1:nrow(GGLsample), nrow(GGLsample), replace = T),]

  df <- sample_boot %>%
  group_by(sex) %>%
  summarise(
    N_Treated = sum(bintreat == 1),
    N_Control = sum(bintreat == 0),
    Mean_Treated = mean(binvote[bintreat == 1]),
    Mean_Control = mean(binvote[bintreat == 0]),
    CATE = Mean_Treated - Mean_Control
  ) %>%
  ungroup()

  diff_boot[i] <- df$CATE[df$sex == "male"]
}

lower.bound <- quantile(diff_boot, 0.025)
upper.bound <- quantile(diff_boot, 0.975)

print(paste0("The Bootstrapped 95% CI is {", lower.bound,", ",upper.bound,"}"))
```

```
[1] "The Bootstrapped 95% CI is {0.0128085564744711, 0.114123356704443}"
```

(c) CATE(female)

```
df <- GGLsample %>%
  group_by(sex) %>%
  summarise(
    N_Treated = sum(bintreat == 1),
    N_Control = sum(bintreat == 0),
    Mean_Treated = mean(binvote[bintreat == 1]),
    Mean_Control = mean(binvote[bintreat == 0]),
    CATE = Mean_Treated - Mean_Control
  ) %>%
  ungroup()

df$CATE[df$sex == "female"]
```

```
[1] 0.09273679
```

This estimator is not casually identified because the treatment is not conditioned on sex.

(d) Bootstrapped CATE(female)

```
set.seed(42)

n <- 10000
diff_boot <- rep(NA, n)

for(i in 1:n){
  sample_boot <-  GGLsample[sample(1:nrow(GGLsample), nrow(GGLsample), replace = T),]

  df <- sample_boot %>%
  group_by(sex) %>%
  summarise(
    N_Treated = sum(bintreat == 1),
```

```
    N_Control = sum(bintreat == 0),
    Mean_Treated = mean(binvote[bintreat == 1]),
    Mean_Control = mean(binvote[bintreat == 0]),
    CATE = Mean_Treated - Mean_Control
  ) %>%
  ungroup()

  diff_boot[i] <- df$CATE[df$sex == "female"]
}
```

```
lower.bound <- quantile(diff_boot, 0.025)
upper.bound <- quantile(diff_boot, 0.975)

print(paste0("The Bootstrapped 95% CI is {", lower.bound,", ",upper.bound,"}"))
```

```
[1] "The Bootstrapped 95% CI is {0.0424547841940011, 0.144271511073702}"
```

(e) Average Treatment Effect (ATE)

```
treated <- mean(GGLsample$binvote[GGLsample$bintreat == 1])
control <- mean(GGLsample$binvote[GGLsample$bintreat == 0])

treated - control
```

```
[1] 0.07741627
```

(f) Bootstrapped ATE

```
set.seed(42)

n <- 10000
diff_boot <- rep(NA, n)

for(i in 1:n){
  sample_boot <- GGLsample[sample(1:nrow(GGLsample), nrow(GGLsample), replace = T),]

  treated <- mean(sample_boot$binvote[sample_boot$bintreat == 1], na.rm = TRUE)
  control <- mean(sample_boot$binvote[sample_boot$bintreat == 0], na.rm = TRUE)

  diff_boot[i] <- treated - control
}
```

```
lower.bound <- quantile(diff_boot, 0.025)
upper.bound <- quantile(diff_boot, 0.975)

print(paste0("The Bootstrapped 95% CI is {", lower.bound,", ",upper.bound,"}"))
```

```
[1] "The Bootstrapped 95% CI is {0.042570988340309, 0.113562788146589}"
```

If we were to repeat this experiment under the same conditions with a sufficiently large sample size, the average treatment effect would fall somewhere within the interval for about 95 out of 100 trials. In this case, we estimate our average treatment effect to fall somewhere in between 0.043 and 0.114.