Problem set 6

Your name here

Due 11/08/2021 at 5pm

NOTE: Start with the file `ps6_2022.Rmd` (available from the github repository at https://github.com/UChicago-pol-methods/IntroQSS-F22/tree/main/assignments). Modify that file to include your answers. Make sure you can "knit" the file (e.g. in RStudio by clicking on the `Knit` button). Submit both the Rmd file and the knitted PDF via Canvas.

In this assignment we will consider data from an experiment that measured the effect of different messages on Michigan residents' likelihood of voting in the August 2006 primary election. The published paper is:

Gerber, Alan S., Donald P. Green, and Christopher W. Larimer. 2008. "Social Pressure and Voter Turnout: Evidence from a Large-Scale Experiment." *American Political Science Review* 102(1): 33-48.

The data file is `ggl.RData` and it is found in the `data` directory of the course github repository.

```
load(url("https://github.com/UChicago-pol-methods/IntroQSS-F22/raw/main/data/ggl.RData"))
```

The dataset will be loaded as an object called `ggl`.

The variables of interest for us in the dataset are as follows:

- `treatment`: which of the treatment did this voter's household receive?
  - "Control": No mailing
  - "CivicDuty": A mailing encouraging voting
- `p2004`: did this voter vote in the primary elections of August 2004? (binary)

We will set a seed, because we'll use some random re-sampling and we would like to see the same results each time we compile this file.

```
set.seed(60637)
```

We will only consider in our analysis households that were assigned the `Control` condition or the `CivicDuty` condition. We will also include only one observation per household. Don't change the code below:

```
ggl <- ggl |>
  filter(treatment == 'Control' | treatment == 'CivicDuty') |>
  group_by(hh_id) |>
  filter(row_number()==1) |>
  ungroup()
```

Question 1: Randomization Inference

Suppose we are only interested in conducting inference over the voters included in the study, and we are interested in treatment effects of the Civic Duty treatment condition relative to control. Treatment was assigned randomly by the researchers under complete random assignment, i.e., they fixed the number of individuals under treatment and under control, and then assigned conditions randomly.

(1a) In the ggl data, create a new variable called `D_civic`, which takes the value 1 if the observation was assigned the CivicDuty condition, and 0 if the observation was assigned

control. Create a new variable called `Y`, which is a copy of `p2004`. Report the number of individuals in treatment (the CivicDuty condition) and control.

```
ggl <- ggl|>
  mutate(D_civic = 1*(treatment == 'CivicDuty'),
         Y = p2004)

ggl |>
  group_by(D_civic) |>
  summarize(n())
```

```
## # A tibble: 2 x 2
##   D_civic `n()`
##     <dbl> <int>
## 1       0 99999
## 2       1 20001
```

(1b) Get the difference-in-means estimate of the ATE on `Y`, and save the estimate as an object called `dm`. Report the value of your difference-in-means estimate of the ATE.

```
dm <- ggl |>
  group_by(D_civic) |>
  summarize(mean = mean(Y)) |>
  pivot_wider(names_from = D_civic, values_from = mean) |>
  summarise(diff = `1` -`0`) |>
    pull(diff)

dm
```

```
## [1] -0.005854747
```

(1c) Create a new column called `newD_civic` which resamples from `D_civic` without replacement. Report the number of individuals assigned treatment and control under `newD_civic`. Is it the same as under the original `D_civic`?

```
ggl <- ggl |>
  mutate(newD_civic = sample(D_civic))


ggl |>
  group_by(newD_civic) |>
  summarize(n())
```

```
## # A tibble: 2 x 2
##   newD_civic `n()`
##        <dbl> <int>
## 1          0 99999
## 2          1 20001
```

(1d) Calculate the difference in means estimate of the average treatment effect UNDER THE RE-SAMPLED TREATMENT, `newD_civic`.

```
ggl |>
  group_by(newD_civic) |>
  summarize(mean = mean(Y)) |>
  pivot_wider(names_from = newD_civic, values_from = mean) |>
  summarise(diff = `1` -`0`) |>
  pull(diff)
```

```
## [1] 0.009144653
```

(1e) Write a randomization inference function that takes a data frame `df` as an argument, then:

- **Creates a new column called `newD_civic` which resamples from D_civic.**
- **Calculates the difference in means estimate of the average treatment effect UNDER THE RE-SAMPLED TREATMENT, `newD_civic`.**
- **Returns the value of estimated ATE as a number (refer to class slides on how to do this).**

Apply your randomization inference function to the `ggl` data and report the estimated ATE.

```
my_ri <- function(df){
  ri_out <- df |>
    # create a new column newD_civic that samples from D_civic
    mutate(newD_civic = sample(D_civic)) |>
    # difference in means estimate under newD_civic
    group_by(newD_civic) |>
    summarize(mean = mean(Y)) |>
    pivot_wider(names_from = newD_civic, values_from = mean) |>
    summarise(diff = `1` -`0`) |>
    pull(diff)

  return(ri_out)
}

my_ri(ggl)
```

```
## [1] -0.003454843
```

(1f) Using `map_dbl()`, apply your function to the `ggl` data 1000 times.

```
null_dist <- map_dbl(1:1000,
                     ~ my_ri(ggl))
```

(1g) Report the proportion of your results from question 1f that have a larger *absolute value* than the *absolute value* of the object `dm`.

```
(pval <- mean(abs(null_dist)>abs(dm)))
```

```
## [1] 0.137
```

How do you interpret the p-value in 1g?

Our procedure a two-sided hypothesis of the sharp null that all individual treatment effects are exactly zero, as compared to the alternative hypothesis that some treatment effects are not zero. The p-value here is the proportion of values of the estimate that we would observe under null that have an absolute magnitude at least as large as the value of the estimate that we observed from the data.

Question 2: Inference from a single random variable

The Poisson distribution is used for counts data. It is sometimes used to model the number of times an event occurs in a given period of time or over a given demographic space. For example, it has been used to model the number of times sectors in London were hit by bombs during World War II.

We will use the `rpois()` function to sample 100 observations from a Poisson distribution, with a mean AND variance of 10, as defined by the "lambda" parameter.

```
y_poisson <- rpois(n = 100, lambda = 10)
```

(2a) Report the theoretical standard deviation of the sample mean if we take a sample of size 100; note that we have said that for this distribution, the variance is 10.

```
sqrt(10/100)
```

```
## [1] 0.3162278
```

(2b) Generate 1000 i.i.d. draws of size 100 from this poisson distribution, and take the sample mean of each of these 1000 draws. Report the standard deviation across these sample means. This is similar to what you did in 1c in pset 5.

This is a simulation that illustrates how the sample mean varies across i.i.d. draws from the same population distribution. For a given sample, we don't generally know what the true reference distribution is, and so we don't get to access this approximation.

```
sim_vec <- map_dbl(1:1000, # for 1000 times
                   # resample w/replacement
                   ~ rpois(n = 100, lambda = 10) |>
                     mean())

sd(sim_vec)
```

```
## [1] 0.3143896
```

(2c) Create an object called `theta_hat` which is the sample mean of `y_poisson`, and an object called `se_hat` that is the estimate of the standard error of the sample mean, using the formula for the unbiased sample variance. Report the values of `theta_hat` and `se_hat`. These are estimates of the mean and the standard error of the sample mean from a single sample.

```
(theta_hat <- mean(y_poisson))
```

```
## [1] 9.6
```

```
(se_hat <- sqrt(var(y_poisson)/length(y_poisson)))
```

```
## [1] 0.3550501
```

(2d) Generate 1000 bootstrapped calculations of the sample mean. Save this object as a vector, and name it `boot_vec`. To do this, take a sample from `y_poisson` of size 100 with replacement 1000 times, and calculate the mean of each of your bootstrapped samples. Report the bootstrapped estimate of the standard deviation of the sample mean.

Refer back to the class notes for more reference on how to do this if you need.

```
boot_vec <- map_dbl(1:1000, # for 1000 times
                    # resample w/replacement
                    ~ sample(y_poisson, replace = TRUE) |>
                      mean()) # and calculate the resampled mean

sd(boot_vec)
```

```
## [1] 0.3504963
```

The bootstrapped estimate is a different estimating strategy than that used in 2c to calculate `theta_hat`, but it is also targeting the standard error of the sample mean, the true value of which is reported in 2a. Bootstrapping is a process that approximates the procedure in part 2b, when we only have a sample and don't know what distributions we are sampling from.

Question 3: Confidence intervals

(3a) The formula for the normal approximation-based confidence intervals is below

$$CI_n = \left(\hat{\theta}_n - z_{1-\alpha/2} \times \hat{\text{se}},\ \hat{\theta}_n + z_{1-\alpha/2} \times \hat{\text{se}}\right)$$

z_c describes the c-th quantile of the standard normal distribution. For 95% confidence intervals, $\alpha = 0.05$, so we want to find $z_{1-\alpha/2} = z_{0.975}$. Using qnorm, get the 97.5-th quantile of the standard normal distribution.

```
(z975 <- qnorm(0.975))
```

```
## [1] 1.959964
```

(3b) Using `theta_hat` and `se_hat` from part 2, and your answer to the previous question, report the 95% normal approximation-based confidence intervals for the estimate of `theta_hat`

```
(CI95 <- c(theta_hat + c(-1,1)*z975*se_hat))
```

```
## [1]  8.904115 10.295885
```

(3c) To get the 90% confidence intervals, we will set α as 0.10. So we want to find $z_{1-\alpha/2} = z_{0.95}$. Using qnorm, get the 95-th quantile of the standard normal distribution.

```
(z95 <- qnorm(0.95))
```

```
## [1] 1.644854
```

(3d) Using `theta_hat` and `se_hat` and your answer from the question above, report the 90% normal approximation-based confidence intervals for the estimate of `theta_hat`.

```
(CI90 <- c(theta_hat + c(-1,1)*z95*se_hat))
```

```
## [1]  9.015994 10.184006
```

(3e) We can compare the distribution of the estimator under the bootstrap procedure and under the normal approximation. Using the `quantile()` function and your saved vector of 1000 bootstrapped estimates of the sample mean, report the 2.5th and 97.5th quantiles of the estimates under the bootstrap. These cover 95% of the empirical distribution of the bootstrap. How do they compare to your 95% normal approximation-based confidence intervals in your answer to 3b above?

```
quantile(boot_vec, probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
##  8.91975 10.27000
```