Problem Set 6: Sampling from a Voter File

Your Name Here

Background

In this exercise, we will focus on sampling and sampling distributions when we have access to an entire census for a given population. In this case, the `data/durham.csv` file contains anonymized data on all registered voters in Durham County, NC that were registered to vote on election day of the 2020 presidential election. The variables in this dataset are:

Name	Description
`ncid`	unique voter identification number
`reg_year`	did person vote (1) or not (0) in 1994?
`age`	age of registered voter
`gender`	voter gender identity (`"F"` = Female, `"M"` = Male, `"U"` = Undeclared)
`race`	voter racial identity (see table below)
`latino`	voter identifies as Hispanic or Latino (`"HL"`), does not (`"NL"`), or undesignated (`"UN"`)
`party`	party registration (`"DEM"` = Democrat, `"Rep"` = Republican, `"LIB"` = Libertarian, `"GRE"` = Green, `"UNA"` = Unaffiliated)
`drivers_lic`	voter has drivers license (1) or not (0)?

North Carolina provides the voter history data in a separate file with a row for each voter who actually casts a ballot in a given election. In the `data/vote_history.csv` file is a row for every voter that cast a ballot in the 2020 general election. The variables in this dataset are:

Name	Description
`ncid`	unique voter identification number
`vote_method_2020`	method of voting in 2020 general election

For the purposed of this exercise, we will treat the voter list from `durhman.csv` as the population of interest. Doing so is an increasingly common approach for polling, where pollsters are now using the voter file as a sampling frame to conduct their polls. We will repeated sample from this population to better understand sampling uncertainty.

Note: please follow the directions carefully about setting the seed for the sampling based questions.

In the Durham data, the racial category codes are:

Race	Description
A	ASIAN
B	BLACK or AFRICAN AMERICAN
I	AMERICAN INDIAN or ALASKA NATIVE
M	TWO or MORE RACES
O	OTHER

Race	Description
P	NATIVE HAWAIIAN or PACIFIC ISLANDER
U	UNDESIGNATED
W	WHITE

Question 1 (6 points)

Load the voter list data (`data/durham.csv`) into R using `read_csv` and save the data as `durham` and use the same function to load in the voter history data (`data/vote_history.csv`) as `vote_history`. Use `left_join` to merge the voter history data into the voter list data so that all rows of the voter list remain. Use the `ncid` variable to join the data sets and save this merged data as `durham`.

Any registered voter who did not turn out in the 2020 general election will not appear in the `vote_history` data so their `vote_method_2020` variable will be missing in the merged data. Create a new variable called `vote_gen_2020` that is 1 if the voter turned out and 0 otherwise (the `is.na()` function may be helpful).

In the write-up, state how many units are in the population (that is, how many rows are in the `durham` data) and the proportion of registered voters that actually voted in the 2020 elections.

Rubric: 1pt for Rmd file compiling (autograder); 3pt for data loaded and merged (autograder), 1pt for creation of the `vote_gen_2020` variable (autograder); 1pt for number of rows and turnout proportion reported (PDF).

Answer 1

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- 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
```

```
durham <- read_csv('data/durham.csv')
```

```
## Rows: 231743 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): ncid, gender, race, latino, party
## dbl (3): reg_year, age, drivers_lic
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
vote_history <- read_csv('data/vote_history.csv')
```

```
## Rows: 178018 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): ncid, vote_method_2020
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
durham <- left_join(durham, vote_history, by = "ncid")
durham <- durham |>
  mutate(vote_gen_2020 = ifelse(is.na(vote_method_2020),0,1))
num_units <- nrow(durham)
prop_voted <- mean(durham$vote_gen_2020)
num_units
```

```
## [1] 231747
```

```r
prop_voted
```

```
## [1] 0.7228702
```

Question 2 (4 points)

Create a density histogram of age with a bin width of 1 and save this plot as `age_hist` (use the aesthetic mapping `y = after_stat(density)` in to accomplish this). Create a barplot for turnout (`vote_gen_2020`) with the proportion on the y-axis (use the aesthetic mapping `y = after_stat(prop)` in `geom_barplot` to achieve this). For a slightly nicer plot, use the `width = 1` argument. Make sure both of these plots are shown in the PDF.

Rubric: 2pts for `age_hist` (autograder); 2pts for `turnout_hist` (autograder).

Answer 2

Question 3 (5 points)

Use `summarize()` to calculate the population mean and standard deviation of `age` and `vote_gen_2020` (that is the mean and SD of these variables in the `durham` data) and save the resulting tibble as `pop_parameters` with the tibble output looking like:

```
# A tibble: 1 × 4
  age_pop_mean age_pop_sd turnout_pop_prop turnout_pop_sd
         <dbl>      <dbl>            <dbl>          <dbl>
1         XX.X       XX.X            X.XXX          X.XXX
```

Make sure that the column names are the same for the autograder. (**Hint:** you can summarize multiple variables in the same call to `summarize`.) Use `knitr::kable()` to present the values in nicely formatted table with `digits = 2` to create nicely rounded numbers and informative column names (they may need to be abbreviated to fit on the page).

Rubric: 4pts for correct `pop_parameters` tibble (autograder); 1pts for nicely formatted table (PDF).

Answer 3

Question 4 (8 points)

In the first line of the code chunk for this question use the following code:

```
library(infer)
set.seed(02138)
```

Then use `rep_slice_sample()` to take 1,000 samples of size 200 from `durham` and calculate the sample mean of `age` and the sample mean/proportion of turnout for each of these samples. Save these as variables named `age_mean` and `turnout_prop` and save the resulting tibble as `samples_n200`.

Create a density histogram of the age means and save this as `age_mean_hist` (use a bin width of 0.5). Create a density histogram of the turnout proportions and save this as `turnout_prop_hist` (use a bin width of 0.005) (**NOTE: this is not a bar plot like in Question 2**). Make sure both of these plots are shown in the PDF and that they have informative labels.

In the write-up, compare the sampling distribution of the sample mean and sample proportion here to the population distributions from question 2. Are the shapes of the sampling distributions similar or different to the population distributions? If different, how are they different?

Rubric: 2pts for correct `sample_n200` output (autograder); 2pts for correct `age_mean_hist` (autograder); 2pts for correct `turnout_prop_hist` (autograder); 1pt informative labels (PDF); 1pt comparison to population distributions (PDF).

Answer 4

Question 5 (7 points)

Use the `summarize()` function on `samples_n200` to calculate the average (named `ev_age` and `ev_turnout`) and standard deviation (named `se_age` and `se_turnout`) of each sample mean/proportion across the repeated samples. Save this tibble as `samp_dist_summary` and it should look like this:

```
# A tibble: 1 × 4
  ev_age se_age ev_turnout se_turnout
   <dbl>  <dbl>      <dbl>      <dbl>
1   X.XX   X.XX      X.XXX      X.XXX
```

Make sure that the column names are the same for the autograder. Use `knitr::kable()` to present the values in nicely formatted table with `digits = 2` to create nicely rounded numbers.

Compare the mean and SD of these sampling distributions to the population means and SDs from the previous question. Are these distributions centered on the same value? Which has more spread, the population distribution of age/turnout or the sampling distributions of their means?

Rubric: 4pts for correct `samp_dist_summary` tibble (autograder); 1pt for nicely formatted table (PDF); 2pts for discussion (PDF).

Answer 5

Question 6 (5 points)

Now suppose we received a bad voter file that only contained voters under the age of 30 at the time of the 2020 election. Set the seed using `set.seed(02138)` and repeat the replicate sampling from Question 3 to take 1,000 samples of size 200 from `durham` after removing voters 30 and over. Calculate the sample proportion of turnout for each of these samples. Save this as a variable named `turnout_prop` and save the resulting tibble as `young_samples_n200`.

Create a density histogram of the turnout proportions with informative labels and add a vertical line at the true population average turnout (of all ages). Save this plot as `young_turnout_prop_hist`.

Does the sampling distribution of turnout in these samples appear biased or unbiased for the population average turnout? Explain why based on the sampling process.

Rubric: 2pts for correct `young_samples_n200` tibble (autograder); 1pt for correctly identifying biased/unbiased (PDF); 2pt explaining bias/unbiasedness (PDF).

Answer 6

Code Printout (DO NOT EDIT BELOW THIS)

```
options(width = 100)
knitr::opts_chunk$set(error = TRUE)
library(tidyverse)
durham <- read_csv('data/durham.csv')
vote_history <- read_csv('data/vote_history.csv')
durham <- left_join(durham, vote_history, by = "ncid")
durham <- durham |>
  mutate(vote_gen_2020 = ifelse(is.na(vote_method_2020),0,1))
num_units <- nrow(durham)
prop_voted <- mean(durham$vote_gen_2020)
num_units
prop_voted
library(infer)
set.seed(02138)
```