STA 322 / 522 Project 1

Will Tirone, Megan Stone

Load Data

We removed counties with 0 population and only work with the 50 U.S. states, since we want to analyze voter data and individuals in these territories can't vote (except D.C.). We also want 2 counties per stratum, and since some of these only have one "county" or county equivalent, we remove them.

We collected our data from the 2010 and 2020 Census at https://data.census.gov/table/, and the voter data we found at https://electionlab.mit.edu/data. Note that we did some pre-processing in Excel to remove some unnecessary data, but the csv data included in our report is what we import here.

Our data consists of the counties listed on the wikipedia page that we extracted using https://wikitable2csv.ggor.de/, Hispanic population data for 2010 and 2020, voting data from the 2020 presidential election, and educational attainment data for the counties.

```
# load data in
counties = read.csv('data/counties.csv')
hispanic_2010 = read.csv('data/new_counties_2010hispanic.csv')
hispanic_2020 = read.csv('data/new_counties_2020hispanic.csv')
education_dat = read.csv('data/education.csv')
voting = read.csv('data/county_votes.csv')
```

Set Up

Now, we'll clean our data and combine it with the county data. This way, we can do less manual scraping work and sample 100 counties, two per state (stratum). The cleaning includes tasks like setting the state and county names to the same format as the other data so we can join on them, combining voter data from different parties, and removing the states and U.S. territories that we don't want to include.

We also had to do some name adjustment in Excel before we loaded the data. We noticed that upon taking the sample, some counties were missing data, but we believed we had the data. There were a variety of causes. For example, in 2022, Connecticut changed their county names! They were updated on the Wikipedia page, but the 2020 census data reflected the old names, so we had to adjust these to the previous names. Alaska also reported their voting data in districts rather than at the county level, so we had to match the districts to Anchorage. We also had some minor name issues like Denver being listed as "Denver, city and county of" which didn't match in the left join on the name "Denver". The matching would be easier with a uniquely identifying number, but unfortunately this wasn't available to us. Note that we only fixed these issues on the 100 counties after the sample was taken, so it's likely that a new sample would contain similar issues.

```
# clean the voting data
voting = voting |>
  group_by(state, county_name, party) |>
  summarise(votes = sum(candidatevotes)) |>
  mutate(state = str_to_title(state),
         county = str_to_title(county_name)) |>
  pivot_wider(names_from = party,
              values_from = c(votes)) |>
  mutate(OTHER = OTHER + GREEN + LIBERTARIAN) |>
  ungroup() |>
  replace_na(list(OTHER = 0)) |>
  mutate(TOTAL = OTHER + REPUBLICAN + DEMOCRAT) |>
  select(state, county, DEMOCRAT, REPUBLICAN, OTHER, TOTAL)
```

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

```
# clean the county data
counties = counties |>
  filter(pop != 0,
         !(state %in% c('Guam', 'District of Columbia',
                        'Northern Mariana Islands',
                        'U.S. Minor Outlying Islands',
                        'Virgin Islands (U.S.)',
                        'American Samoa',
                        'Puerto Rico'))) |>
  mutate(density = pop / area)

# clean the hispanic pop data
```

```r
hispanic_data = left_join(hispanic_2020, hispanic_2010,
                          by = c('state', 'county')) |>
  rename(h_pop_2020 = hispanic_pop.x,
         h_pop_2010 = hispanic_pop.y) |>
  mutate(change = h_pop_2020 - h_pop_2010)

# and combine all the data
counties = left_join(counties, voting, by = c('state', 'county'))
counties = left_join(counties, hispanic_data, by = c('state', 'county'))
counties = left_join(counties, education_dat, by = c('state', 'county'))
```

Additional Cleaning Notes

We completed additional cleaning steps in a separate R script and imported the csv data shown above. The code below is not run in this notebook as the files were exported and then imported above. We include this code for completeness.

To make the "geographic" variable match with the one from the wikipedia table (and therefore join together seamlessly), we cleaned the data by dissecting the county name from the state name and creating two separate variables labeled "county" and "state." Of note, we had to dissect on "Parish" for Louisiana, "Borough" or "Census" area for Alaska, and "city" for select counties in Virginia. All of the cleaning was performed in a separate R Script, exported to a CSV, and then imported into our analysis script for processing.

```r
read_csv('counties_hispanic.csv')

counties_hispanic = counties_hispanic %>%
  group_by(County) %>%
  mutate(comma_idx = unlist(gregexpr(',', County))[1]) %>%
  mutate(county_idx = unlist(gregexpr('County', County))[1])

counties_hispanic$State = substr(counties_hispanic$County,
                                 counties_hispanic$comma_idx+2,
                                 nchar(counties_hispanic$County))

counties_hispanic$new_county = substr(counties_hispanic$County, 1,
                                      counties_hispanic$county_idx-2)

# fixing Alaska
Alaska = counties_hispanic[((counties_hispanic$State == "Alaska") == TRUE),]
```

```
Alaska$new_county = substr(Alaska$County, 1, Alaska$comma_idx-1)
counties_hispanic$new_county[((counties_hispanic$State == "Alaska")
                              == TRUE)] = Alaska$new_county

# fixing Louisiana
Louisiana = counties_hispanic[((counties_hispanic$State == "Louisiana")
                               == TRUE),]
Louisiana$new_county = substr(Louisiana$County, 1, Louisiana$comma_idx-1)
counties_hispanic$new_county[((counties_hispanic$State == "Louisiana")
                              == TRUE)]
= Louisiana$new_county

# fixing Virginia
Virginia_weird = counties_hispanic[((counties_hispanic$State == "Virginia")
                                    == TRUE &
                                     (counties_hispanic$county_idx == -1)
                                    == TRUE),]
Virginia_weird$new_county = substr(Virginia_weird$County, 1,
                                   Virginia_weird$comma_idx-6)

counties_hispanic$new_county[((counties_hispanic$State == "Virginia") == TRUE &
                             (counties_hispanic$county_idx == -1)
                             == TRUE)] = Virginia_weird$new_county

for (i in nrow(Virginia)){
if (Virginia$county_idx[i] == -1){
  Virginia$new_county[i] = substr(Virginia$County[i], 1,
                                  Virginia$comma_idx[i]-5)
}}

final_dataset = data.frame(county = counties_hispanic$new_county,
                           state = counties_hispanic$State,
                           hispanic_pop = counties_hispanic$HispanicPop)

final_dataset = final_dataset[1:3143,]
```

Collect Sample

Since we want to capture variability across the whole U.S., it seems like a fair approach to stratify across the states. But, we want to sample large population centers with a higher

probability than rural areas with low population. By stratifying, we'll reduce variance than if we cluster sampled or used a simple random sample. To do that, we'll use a pps sample:

1. Stratify by state, H = 50.
2. Select 2 counties in each state by pps sample, using $\pi_i = n_h \frac{x_i}{t_{x_h}} = 2\frac{\text{county pop}}{\text{state pop}}$, so our weights are then $1/\pi_i$.

```
# initialize
set.seed(1822)
sampled_counties = data.frame()
states = unique(counties$state)

for (s in states) {

  # select a single state as a subset of the data
  subset = counties |> filter(state == s)

  # create weights
  subset = subset |>
    mutate(wt = 1 / (2 * ( pop / sum(pop))))

  # draw the indices that correspond to counties
  county_index = ppss(subset$pop, 2)

  # adding rows to dataframe by index matching to state + county
  sampled_counties = rbind(sampled_counties,
                           data.frame(subset[county_index, ]))
}
```

Q1)

Estimating average population density per county below.

Note that our estimate is somewhat far off the truth, which is 93.8 residents per sq. mile per https://www.census.gov/data/tables/time-series/dec/density-data-text.html, though our confidence interval does include this. Of course, we've mostly sampled large population centers which will have a higher population density than average. If we wanted to correct for this, we might conduct an SRS within each state and select 5-10 counties.

```
design = svydesign(~1, strata = ~state, weights = ~wt,
                   data = sampled_counties)
```

```r
dens = svymean(~density, design)

cat("Avg. pop. density per county (sq. mi.) in the US, 2020 : ",
    '\n', dens, '\n', '95% C.I. : \n',  confint(dens))
```

```
Avg. pop. density per county (sq. mi.) in the US, 2020 :
 476.9063
 95% C.I. :
 68.99317 884.8195
```

Q2)

We'll continue to use the same design and estimate the total number of people in the U.S. who identify as Hispanic or Latino, any race, below. According to the census, the total population was 62,080,044 in 2020 and we're decently close. Again, the estimate is within our confidence interval.

```r
h_2020 = svytotal(~h_pop_2020, design)

cat("Total Hispanic + Latino population in US, 2020 : ",
    '\n', h_2020, '\n', '95% C.I. : \n',  confint(h_2020))
```

```
Total Hispanic + Latino population in US, 2020 :
 72890593
 95% C.I. :
 59377466 86403720
```

Q3)

Again using the same approach with our `change` variable, which is the population in 2020 that identify as Hispanic or Latino minus the number in 2010. Again comparing to Census.gov, the U.S. had 50,477,594 in 2010 and 62,080,044 in 2020, so we have a very good estimate here.

```r
change = svytotal(~change, design)

cat("Total change in Hispanic + Latino population in US, 2010 to 2020 : ",
    '\n', change, '\n', '95% C.I. : \n',  confint(change))
```

```
Total change in Hispanic + Latino population in US, 2010 to 2020 :
 12268809
 95% C.I. :
 8575427 15962191
```

Q4)

To calculate the percentage of voters of each party in 2020, we'll estimate the total number of voters, and using that point estimate, calculate the percentages and confidence intervals for each party. The total voter estimate is 154,075,798 with the actual number of voters at 155,507,476 per the census. Again, a very close estimate. We have to do a bit of manual calculation here, though. And according to the Federal Election Commission, the percentage of Republican voters was 46.8% and Democrats at 51.3%. Our estimates look pretty good!

```
TOTAL = svytotal(~TOTAL, design)[1]
R = svytotal(~REPUBLICAN, design)
D = svytotal(~DEMOCRAT, design)
O = svytotal(~OTHER, design)

R_point_est = R[1] / TOTAL
R_conf_int = confint(R) / TOTAL
cat('Percent of Republican Voters, 2020 Election : \n', R_point_est,
    '\n', '95% C.I. : \n', R_conf_int, '\n \n')
```

```
Percent of Republican Voters, 2020 Election :
 0.480865
 95% C.I. :
 0.4289064 0.5328236
```

```
D_point_est = D[1] / TOTAL
D_conf_int = confint(D) / TOTAL
cat('Percent of Democrat Voters, 2020 Election : \n', D_point_est,
    '\n', '95% C.I. : \n', D_conf_int, '\n \n')
```

```
Percent of Democrat Voters, 2020 Election :
 0.5051539
 95% C.I. :
 0.4511108 0.559197
```

```r
O_point_est = O[1] / TOTAL
O_conf_int = confint(O) / TOTAL
cat('Percent of Other Voters, 2020 Election : \n', O_point_est,
    '\n', '95% C.I. : \n', O_conf_int)
```

```
Percent of Other Voters, 2020 Election :
 0.01398113
 95% C.I. :
 0.01272578 0.01523647
```

Q5)

We scraped county-level educational attainment data from the Census as well, and we want to know the total number of individuals with a Bachelor's degree or higher. Per the Census this is 35.6% of the population, which is 117,995,944 in 2020. We're very close again!

```r
bach = svytotal(~Bachelors_and_Higher, design)

cat("Total Number of Individuals With a Bachelors or Higher, 2020 : ",
    '\n', bach, '\n', '95% C.I. : \n',  confint(bach))
```

```
Total Number of Individuals With a Bachelors or Higher, 2020 :
 118726046
 95% C.I. :
 107279712 130172381
```