Project 1

Elyse McFalls and Holly Cui

2023-10-20

Data and Sampling Frame

```
county <- read_excel("/Users/hollycui/Desktop/STA\ 522/Project1/county.xlsx",
                     col_names = TRUE)
```

This project seeks to understand facets of the United States using county-level data. Specifically, we are interested in examining the characteristics of counties in the U.S., such as population density, demographic information, and political preferences. Data on the states and counties in the United States were collected from the Wikipedia page List of states and territories of the United States which contains the 2020 census estimates for each county's population and area. More detailed data for each county was extracted from the Unites States Census Bureau and MIT's Election Data and Science Labs (MEDSL). The data from the Census contained demographic estimates for each county from 2010 to 2020. MEDSL on the other hand, entailed data on political participation by county from 2000 to 2020.

To perform the survey, all counties from the 50 U.S. states and the District of Columbia were included in the sampling frame. However, counties from U.S. territories and remote islands were not incorporated due to the fact that these regions are often located far from the continental United States, making data collection logistically challenging and costly from a realistic point of view. Even if we are presented with the data estimates of all counties, U.S. territories and remote islands tend to have smaller populations compared to states, which can result in less statistical significance when conducting a national survey. Including these areas may not significantly impact the overall survey results. Finally, to align with one of our objectives on the topic of political preferences, we are primarily interested in regions that are enfranchised or enabled representation in the presidential votes. Since residents of these regions, such as Puerto Rico, Guam, or American Samoa, do not have full voting representation in the general elections, we decide to not consider them as part of the sampling frame for our study.

```
# Remove US territories & remote islands
US_territories <- c("American Samoa", "Guam", "Northern Mariana Islands",
                    "Puerto Rico", "U.S. Minor Outlying Islands",
                    "Virgin Islands (U.S.)")

clean_county <- county %>%
  filter(!state %in% US_territories) %>%
  group_by(state) %>%
  mutate(Nh = n(), Sh = sd(pop)) %>%
  ungroup() %>%
  mutate(id = row_number())
```

Sampling Procedure

Sampling Design

In deciding on the sampling design, we wanted to choose a framework that best fit the goals of this project. This framework would ideally give us a representative sample of the counties in the U.S. that bears in mind the variedness of each state. Therefore, we decided to conduct a simple stratified sampling with each of the 50 states serving as individual stratum. To determine the sample size in each stratum, we adopted optimal allocation to minimize the variance of estimates. In terms of the District of Columbia which operates simultaneously as a city, a county and a state, we treated it as a singleton stratum where the probability of DC being sampled equals to 1. This can be further analyzed as a certainty sample with detailed treatments being explained in the estimation section.

By taking a stratified sample, we are ensuring that every state in the U.S. is being accounted for. Stratified samples also provide estimates with lower standard errors when there is more between-variance than within-variance. We expect this to be the case with counties in each state. For instance, counties in Virginia are expected to exhibit more similarities among themselves than when compared to counties in Wyoming. However, we also believed that the variance of our variables of interest within each state may differ by state. Consider a state like North Dakota where most of the counties are rural compared to a state like North Carolina that has a mixture of rural and urban areas. We'd anticipate that county-level population in North Dakota would be more uniform and the same statistics in North Carolina would be more varied. Therefore, instead of sampling proportional to size, we opted for an optimal allocation framework (Eq. 1). This way, we would sample more from states that have higher variances for our variables of interest. Specifically, we decided to focus on the population variable of each county to optimally allocate the samples of each stratum since we expect population values to be related to demographics. Therefore, this framework will subsequently reduce the variance in our population estimates and likely the variance in our demographic related estimates.

$$\text{Eq. } 1 : n_h = n \times \frac{N_h S_h / N}{\sum_{h=1}^{H} N_h S_h / N}$$

where:

- n_h is the number of sample in state h
- N_h is the number of counties in state h
- N is the total number of counties in the sampling frame
- n is the sample size
- H is the total number of states
- S_h is the variance of the county populations in state h

Furthermore, employing a simple random sampling approach following stratification enables us to obtain a representative sample from each state. While we contemplated adopting a sample proportional to size, we ultimately decided against it as we deemed it unnecessary to give higher priority to more densely populated areas.

Additionally, we made the deliberate choice to include the District of Columbia (D.C.) in our sample with 100% certainty. This decision stems from D.C.'s unique status as the nation's capital, with characteristics akin to both a state and a county. If we were to treat D.C. as a state with a single county, our optimal allocation formula would assign zero samples to it, given the absence of variance in that context. Therefore, to ensure the inclusion of D.C. in our sample, we intentionally retained it as a separate certainty PSU in our sampling strategy.

```r
# State-level strata statistics (without DC - certainty PSU)
clean_state <- county %>%
  filter(!state %in% US_territories) %>%
  filter(state != "District of Columbia") %>%
  group_by(state) %>%
  summarise(Nh = n(), Sh = sd(pop)) %>%
  ungroup()
```

Sample Sizes and Weights

Sample Size (n): Initially, we selected a sample size of 314, equivalent to 10% of our population size (which is usually a good maximum sample size). However, when applying the optimal allocation formula, it produced instances where certain states had no counties included in the sample. To ensure representation from every state, we rounded any calculated values of nh less than 0.5 up to 1. As a result, our final sample size was adjusted to 317.

```r
# Calculate denominator for optimal allocation
n = 314
denominator = sum(clean_state$Nh * clean_state$Sh)

# Summarize nh for each state stratum
state_strata <- clean_state %>%
  mutate(nh_round = round((n-1)*Nh*Sh / denominator)) %>%
  # Check rounding issue
  mutate(nh = ifelse(nh_round == 0, nh_round+1, nh_round))

# Fix rounding & get final sampling schema
state_nh <- state_strata %>%
  select(state, Nh, Sh, nh)
```

```r
# Sampling
set.seed(123)
sample_county = c()
for (i in 1:nrow(state_nh)) {
  sample_by_state = sample(clean_county$id[clean_county$state == state_nh$state[i]],
                           state_nh$nh[i])
  sample_county = c(sample_county, sample_by_state)
}

# Add back DC (certainty PSU)
final_sample = c(sample_county, 322)

# Final sample
sample = clean_county %>%
  filter(id %in% final_sample)
```

Weights: The weight assigned to each county is determined by dividing the total number of counties within its corresponding state by the number of counties actually sampled. For example, in the case of Alabama, which has a total of 67 counties and of which 3 were included in the sample, the weight assigned to Alabama's sampled county is calculated as 67 / 3, resulting in a weight of approximately 22.33. In contrast, the District of Columbia (D.C.) carries a weight of 1 since it serves as a primary sampling unit with 100% certainty.

```r
# Excluding D.C. for now, its weight is 1
sample_final <- sample %>%
  left_join(state_nh, by = c("state", "Nh", "Sh")) %>%
  select(-id) %>%
  mutate(weights = Nh/nh,
         pop_density = pop/area,
         # Format city's name
         county = ifelse(grepl("city", county, ignore.case = TRUE),
                         sub(",.*", "", county),
                         county))

#write_xlsx(sample_final, "/Users/hollycui/Desktop/STA\ 522/Project1/sample_final.xlsx")
```

The breakdown of the number of counties sampled by state using the optimal allocation formula is as follows:

State	nh	State	nh	State	nh
Alabama	3	Louisiana	3	Ohio	9
Alaska	1	Maine	1	Oklahoma	4
Arizona	8	Maryland	3	Oregon	3
Arkansas	2	Massachusetts	4	Pennsylvania	9
California	39	Michigan	10	Rhode Island	1
Colorado	5	Minnesota	6	South Carolina	3
Connecticut	1	Mississippi	2	South Dakota	1
Delaware	1	Missouri	7	Tennessee	6
Florida	16	Montana	1	Texas	48
Georgia	11	Nebraska	3	Utah	3
Hawaii	1	Nevada	4	Vermont	1
Idaho	2	New Hampshire	1	Virginia	8
Illinois	25	New Jersey	3	Washington	7
Indiana	5	New Mexico	2	West Virginia	1
Iowa	3	New York	16	Wisconsin	5
Kansas	4	North Carolina	8	Wyoming	1
Kentucky	4	North Dakota	1	District of Columbia*	1

- *certainty primary sampling unit*

County-Level Data Collection

Building upon our sample of 317 counties, our next step is to acquire more detailed, county-level data. This data is essential for addressing the specific questions at hand, particularly those related to county population density, demographic compositions (such as the Hispanic or Latino population), and political affiliations. Therefore, the upcoming data collection efforts are carefully aligned with our objectives and will draw from reliable, professional sources.

Political Preferences

County Presidential Election Returns 2000-2020 is an open-access dataset managed by MIT's Election Data and Science Labs (MEDSL). This dataset provides comprehensive records of county-level presidential election statistics spanning from the year 2000 to 2020. Each entry in the dataset corresponds to a specific county and election year, featuring the candidates of various political parties (Democrat, Republican, Green, Libertarian, and Other), along with their vote counts and the total number of votes cast in that county. Given our specific interest in the 2020 election, we have filtered out data from other years to focus the 2020 election cycle. To align this dataset with our sample dataset, we treat all parties other than Democrat or Republican as third party, and group by county and party to retrieve the number of votes for each party. Additionally, we have included the total number of votes cast in each county, enabling us to further analyze and draw insights from the 2020 election data.

Before proceeding with the integration of the retrieved values into our sample dataset, it's essential to acknowledge the considerable variability in county names across datasets. To illustrate, one of our sampled counties, *Northwest Hills Planning Region* in Connecticut, has undergone a name change from *Litchfield*. However, a majority of professional sources online still maintain the previous naming convention. As such, relying solely on a brute-force matching method may not yield accurate results. After thorough online research, we have identified a conventional solution: the use of 5-digit Federal Information Processing Series (FIPS) codes to standardize county names. Since our data source also includes FIPS information, our strategy involves initially establishing a match between our sampled county names and a FIPS dataset. Subsequently, we will perform the data integration, linking the party data based on the FIPS codes. This approach significantly enhances our efficiency compared to the manual verification of county names.

```
# Use FIPS for matching
fips <- read.csv("/Users/hollycui/Desktop/STA\ 522/Project1/countyfipstool20190120.csv")

most_match <- sample_final %>%
  left_join(fips, by = join_by("county" == "cname", "state" == "sname")) %>%
  select(-c(sab, sid, sfips, saint, cfips))

# Manually impute missing values
# write_xlsx(most_match, "/Users/hollycui/Desktop/STA\ 522/Project1/sample_match.xlsx")
```

```
# Read-in fixed sample with complete FIPS
sample_match <- read_excel("/Users/hollycui/Desktop/STA\ 522/Project1/sample_match.xlsx",
                           col_names = TRUE)

# Read-in vote dataset
vote <- read.csv("/Users/hollycui/Desktop/STA\ 522/Project1/countypres_2000-2020.csv")
```

```
# Create vote dataframe by party and county
third_party <- c("OTHER", "GREEN", "LIBERTARIAN")

vote_by_party <- vote %>%
  filter(year == 2020) %>%
  select(state, state_po, county_name, county_fips,
         candidate, party, candidatevotes, totalvotes) %>%
  mutate(party_group = case_when(
    party %in% third_party ~ "THIRD",
    .default = party
  )) %>%
  select(-c(party, candidate)) %>%
  group_by(state, county_name, party_group) %>%
  summarise(votes = sum(candidatevotes), .groups = "drop")

# Clean original data
vote_select <- vote %>%
  filter(year == 2020) %>%
```

```r
  mutate(party_group = case_when(
    party %in% third_party ~ "THIRD",
    .default = party)) %>%
  select(state, state_po, county_name, county_fips, party_group, totalvotes) %>%
  distinct() # remove duplicate rows

# Join back for final vote data
clean_vote <- vote_by_party %>%
  left_join(vote_select, by = join_by(state, county_name, party_group)) %>%
  select(state, state_po, county_name, county_fips, party_group, votes, totalvotes) %>%
  pivot_wider(names_from = party_group, values_from = votes) %>%
  mutate(county_fips = ifelse(state_po == "DC", 11001, county_fips))
```

Following our final attempt to perform the match-back, we encountered an issue with *Wrangell* in Alaska, which was not documented in the vote data. To address this discrepancy, we turned to the official data source, specifically the table available on page 2 of Alaska's government website, for a manual imputation process. This step was necessary to ensure the completeness and accuracy of our dataset, particularly for the county of Wrangell in Alaska.

```r
# Match vote with sample data
sample_w_vote <- sample_match %>%
  left_join(clean_vote, by = join_by("fips" == "county_fips")) %>%
  # Manual fix Wrangell data based on local government result
  mutate(DEMOCRAT = ifelse(county == "Wrangell", 171, DEMOCRAT),
         REPUBLICAN = ifelse(county == "Wrangell", 526, REPUBLICAN),
         THIRD = ifelse(county == "Wrangell", 33, THIRD),
         totalvotes = ifelse(county == "Wrangell", 730, totalvotes)) %>%
  select(-c(state.y, state_po, county_name)) %>%
  rename(state = state.x)
```

The final matched sample dataset incorporates new information as follows:

- `totalvotes` is the total number of valid votes in the county
- `DEMOCRAT` is the number of votes to Democrat candidates in the county
- `REPUBLICAN` is the number of votes to Republican candidates in the county
- `THIRD` is the number of votes to third party candidates in the county

Hispanic / Latino Population

- 2020

```r
# estimates 7/1/2020
hispanic_data_2020 <-
  read.csv('/Users/hollycui/Desktop/STA\ 522/Project1/demo_2010_2020.csv') %>%
  filter(YEAR == 13, AGEGRP == 0) %>%
  select(STNAME, CTYNAME, H_MALE, H_FEMALE)
```

```r
# remove "County" from names
# weird ch in new mexico county name, not in sample
hispanic_data_2020 <- hispanic_data_2020[-c(1804),]
hispanic_data_2020$CTYNAME <- sub('County', '', hispanic_data_2020$CTYNAME)
```

```r
sample_st_cty <- (sample %>% mutate(st_cty = paste(state, county, sep="")))$st_cty

# keep counties in the sample
hispanic_sample_2020 <- hispanic_data_2020 %>%
  mutate(CTYNAME = trimws(CTYNAME, which = "right"),
         n_hispanic = as.numeric(H_MALE) + as.numeric(H_FEMALE),
         st_cty = paste(STNAME, CTYNAME, sep="")) %>%
  filter(st_cty %in% sample_st_cty | CTYNAME %in% c('Hawaii', 'San Francisco',
                                                    'Wrangell City and Borough',
                                                    'Emporia city', 'Salem city',
                                                    'Radford city', 'De Witt',
                                                    'Litchfield')) %>%
  # rename county names
  mutate(CTYNAME = ifelse(CTYNAME == "Wrangell City and Borough", "Wrangell", CTYNAME),
         CTYNAME = ifelse(CTYNAME == "De Witt", "DeWitt", CTYNAME),
         CTYNAME = ifelse(CTYNAME == "Emporia city", "Emporia", CTYNAME),
         CTYNAME = ifelse(CTYNAME == "Radford city", "Radford", CTYNAME),
         CTYNAME = ifelse(CTYNAME == "Salem city", "Salem", CTYNAME))
```

```r
sample_w_hisp2020 <- sample_w_vote %>%
  mutate(state = ifelse(county == "Hawaii", "Hawaii", state)) %>%
  left_join(hispanic_sample_2020, by = join_by("state" == "STNAME",
                                               "county" == "CTYNAME")) %>%
  select(-c(H_MALE, H_FEMALE, st_cty)) %>%
  rename(hisp_2020 = n_hispanic) %>%
  relocate(hisp_2020, .after = "pop_density")
```

- 2010

```r
# estimates 7/1/2010
hispanic_data_2010 <- read.csv('demo_2010_2020.csv') %>%
  filter(YEAR == 3, AGEGRP == 0) %>%
  select(STNAME, CTYNAME, H_MALE, H_FEMALE)

head(hispanic_data_2010)
```

```
##    STNAME        CTYNAME H_MALE H_FEMALE
## 1 Alabama Autauga County    690      619
## 2 Alabama Baldwin County   4448     3612
## 3 Alabama Barbour County    747      597
## 4 Alabama    Bibb County    301      110
## 5 Alabama  Blount County   2584     2081
## 6 Alabama Bullock County    429      346
```

```r
# remove county from names
# weird ch in new mexico county name, not in sample
hispanic_data_2010 <- hispanic_data_2010[-c(1804),]
hispanic_data_2010$CTYNAME <- sub('County', '', hispanic_data_2010$CTYNAME)
```

```r
# keep counties in the sample
hispanic_sample_2010 <- hispanic_data_2010 %>%
```

```r
  mutate(CTYNAME = trimws(CTYNAME, which = "right"),
         n_hispanic = as.numeric(H_MALE) + as.numeric(H_FEMALE),
         st_cty = paste(STNAME, CTYNAME, sep="")) %>%
  filter(st_cty %in% sample_st_cty | CTYNAME %in% c('Hawaii', 'San Francisco',
                                                    'Wrangell City and Borough',
                                                    'Emporia city', 'Salem city',
                                                    'Radford city', 'De Witt',
                                                    'Litchfield')) %>%
    # rename county names
  mutate(CTYNAME = ifelse(CTYNAME == "Wrangell City and Borough", "Wrangell", CTYNAME),
         CTYNAME = ifelse(CTYNAME == "De Witt", "DeWitt", CTYNAME),
         CTYNAME = ifelse(CTYNAME == "Emporia city", "Emporia", CTYNAME),
         CTYNAME = ifelse(CTYNAME == "Radford city", "Radford", CTYNAME),
         CTYNAME = ifelse(CTYNAME == "Salem city", "Salem", CTYNAME))
```

```r
sample_w_hisp2010 <- sample_w_hisp2020 %>%
  left_join(hispanic_sample_2010, by = join_by("state" == "STNAME",
                                               "county" == "CTYNAME")) %>%
  select(-c(H_MALE, H_FEMALE, st_cty)) %>%
  rename(hisp_2010 = n_hispanic) %>%
  relocate(hisp_2010, .after = "hisp_2020")
```

- Q5 exploration: Aging of population (estimate proportion of age 65+ in 2020 and 2010)
  - Get 2010 population data by county:

```r
pop_2010 <- read.csv('demo_2010_2020.csv') %>%
  filter(YEAR == 3 & AGEGRP == 0) %>%
  select(STNAME, CTYNAME, TOT_POP)

head(pop_2010)
```

```
##    STNAME        CTYNAME TOT_POP
## 1 Alabama Autauga County   54761
## 2 Alabama Baldwin County  183121
## 3 Alabama Barbour County   27325
## 4 Alabama    Bibb County   22858
## 5 Alabama  Blount County   57372
## 6 Alabama Bullock County   10876
```

```r
# remove county from names
# weird ch in new mexico county name, not in sample
pop_2010 <- pop_2010[-1804,]
pop_2010$CTYNAME <- sub('County', '', pop_2010$CTYNAME)
```

```r
# keep counties in the sample
pop_sample_2010 <- pop_2010 %>%
  mutate(CTYNAME = trimws(CTYNAME, which = "right"),
         pop.2010 = as.numeric(TOT_POP),
         st_cty = paste(STNAME, CTYNAME, sep="")) %>%
  filter(st_cty %in% sample_st_cty | CTYNAME %in% c('Hawaii', 'San Francisco',
```

```
                                                    'Wrangell City and Borough',
                                                    'Emporia city', 'Salem city',
                                                    'Radford city', 'De Witt',
                                                    'Litchfield')) %>%
    # rename county names
  mutate(CTYNAME = ifelse(CTYNAME == "Wrangell City and Borough", "Wrangell", CTYNAME),
         CTYNAME = ifelse(CTYNAME == "De Witt", "DeWitt", CTYNAME),
         CTYNAME = ifelse(CTYNAME == "Emporia city", "Emporia", CTYNAME),
         CTYNAME = ifelse(CTYNAME == "Radford city", "Radford", CTYNAME),
         CTYNAME = ifelse(CTYNAME == "Salem city", "Salem", CTYNAME))

# join back
sample_w_pop2010 <- sample_w_hisp2010 %>%
  left_join(pop_sample_2010, by = join_by("state" == "STNAME",
                                          "county" == "CTYNAME")) %>%
  select(-c(TOT_POP, st_cty)) %>%
  relocate(pop.2010, .after = "pop")
```

```
- Get 65+ age group by county:
```

```
age_65 <- read.csv('demo_2010_2020.csv') %>%
  filter(YEAR %in% c(3, 13) & AGEGRP >= 14) %>%
  select(STNAME, CTYNAME, YEAR, AGEGRP, TOT_POP)

head(age_65)
```

```
##    STNAME        CTYNAME YEAR AGEGRP TOT_POP
## 1 Alabama Autauga County    3     14    2289
## 2 Alabama Autauga County    3     15    1752
## 3 Alabama Autauga County    3     16    1259
## 4 Alabama Autauga County    3     17     743
## 5 Alabama Autauga County    3     18     556
## 6 Alabama Autauga County   13     14    2806
```

```
# remove county from names
# weird ch in new mexico county name, not in sample
age_65 <- age_65[-c(18031:18040),]
age_65$CTYNAME <- sub('County', '', age_65$CTYNAME)
```

```
age_sample_65 <- age_65 %>%
  mutate(CTYNAME = trimws(CTYNAME, which = "right"),
         TOT_POP = as.numeric(TOT_POP)) %>%
  group_by(STNAME, CTYNAME, YEAR) %>%
  summarise(pop.65 = sum(TOT_POP), .groups = "drop") %>%
  mutate(YEAR = ifelse(YEAR == 3, "pop_2010_65", "pop_2020_65")) %>%
  pivot_wider(names_from = YEAR, values_from = pop.65) %>%
  mutate(st_cty = paste(STNAME, CTYNAME, sep="")) %>%
  filter(st_cty %in% sample_st_cty | CTYNAME %in% c('Hawaii', 'San Francisco',
                                                    'Wrangell City and Borough',
                                                    'Emporia city', 'Salem city',
                                                    'Radford city', 'De Witt',
                                                    'Litchfield')) %>%
```

```r
  mutate(CTYNAME = ifelse(CTYNAME == "Wrangell City and Borough", "Wrangell", CTYNAME),
         CTYNAME = ifelse(CTYNAME == "De Witt", "DeWitt", CTYNAME),
         CTYNAME = ifelse(CTYNAME == "Emporia city", "Emporia", CTYNAME),
         CTYNAME = ifelse(CTYNAME == "Radford city", "Radford", CTYNAME),
         CTYNAME = ifelse(CTYNAME == "Salem city", "Salem", CTYNAME))

# join back
sample_complete <- sample_w_pop2010 %>%
  left_join(age_sample_65, by = join_by("state" == "STNAME",
                                        "county" == "CTYNAME")) %>%
  select(-c(st_cty, fips)) %>%
  relocate(pop_2020_65, .after = "THIRD") %>%
  rename(pop.2020 = pop)
```

```r
# write.csv(sample_complete,
#           file = "/Users/hollycui/Desktop/STA\ 522/Project1/sample_complete.csv",
#           row.names = FALSE)
```

Estimations

```r
sample_complete <-
  read.csv(file = "/Users/hollycui/Desktop/STA\ 522/Project1/sample_complete.csv")
# DC certainty PSU
sample_complete[61, 9] = 1
```

```r
# survey design
options(survey.lonely.psu="certainty")
des <- svydesign(~1,
                 strata = sample_complete$state,
                 weights = sample_complete$weights,
                 fpc = sample_complete$Nh,
                 data = sample_complete)
```

- Q1

```r
svymean(~pop_density, des)
```

```
##               mean     SE
## pop_density 180.84 29.677
```

```r
confint(svymean(~pop_density, des))
```

```
##                2.5 %   97.5 %
## pop_density 122.6699 239.0017
```

- Q2

```r
svytotal(~hisp_2020, des)
```

```
##                total       SE
## hisp_2020 61180804 10410749
```

```r
confint(svytotal(~hisp_2020, des))
```

```
##               2.5 %   97.5 %
## hisp_2020 40776112 81585496
```

- Q3

```r
# Method 1
sample_change <- sample_complete %>%
  mutate(change = hisp_2020 - hisp_2010)

des1 <- svydesign(~1,
                 strata = sample_change$state,
                 weights = sample_change$weights,
                 fpc = sample_change$Nh,
                 data = sample_change)

svytotal(~change, des1)
```

```
##            total      SE
## change 10391602 1603186
```

```r
confint(svytotal(~change, des1))
```

```
##          2.5 %   97.5 %
## change 7249416 13533788
```

- Q4

```r
svyratio(~REPUBLICAN, ~totalvotes, des)
```

```
## Ratio estimator: svyratio.survey.design2(~REPUBLICAN, ~totalvotes, des)
## Ratios=
##            totalvotes
## REPUBLICAN  0.4839342
## SEs=
##            totalvotes
## REPUBLICAN 0.01721629
```

```r
confint(svyratio(~REPUBLICAN, ~totalvotes, des))
```

```
##                            2.5 %    97.5 %
## REPUBLICAN/totalvotes 0.4501909 0.5176775
```

```r
svyratio(~DEMOCRAT, ~totalvotes, des)
```

```
## Ratio estimator: svyratio.survey.design2(~DEMOCRAT, ~totalvotes, des)
## Ratios=
##          totalvotes
## DEMOCRAT  0.4991369
## SEs=
##          totalvotes
## DEMOCRAT 0.01747804
```

```r
confint(svyratio(~DEMOCRAT, ~totalvotes, des))
```

```
##                          2.5 %    97.5 %
## DEMOCRAT/totalvotes 0.4648806 0.5333932
```

```r
svyratio(~THIRD, ~totalvotes, des)
```

```
## Ratio estimator: svyratio.survey.design2(~THIRD, ~totalvotes, des)
## Ratios=
##       totalvotes
## THIRD 0.01692889
## SEs=
##         totalvotes
## THIRD 0.0006317843
```

```r
confint(svyratio(~THIRD, ~totalvotes, des))
```

```
##                        2.5 %     97.5 %
## THIRD/totalvotes 0.01569061 0.01816716
```

- Q5

```r
# Proportion of US population age 65+ in 2020
svyratio(~pop_2020_65, ~pop.2020, des)
```

```
## Ratio estimator: svyratio.survey.design2(~pop_2020_65, ~pop.2020, des)
## Ratios=
##               pop.2020
## pop_2020_65 0.1682854
## SEs=
##                 pop.2020
## pop_2020_65 0.003539639
```

```r
confint(svyratio(~pop_2020_65, ~pop.2020, des))
```

```
##                          2.5 %   97.5 %
## pop_2020_65/pop.2020 0.1613478 0.175223
```

```r
# Proportion of US population age 65+ in 2010
svyratio(~pop_2010_65, ~pop.2010, des)
```

```
## Ratio estimator: svyratio.survey.design2(~pop_2010_65, ~pop.2010, des)
## Ratios=
##              pop.2010
## pop_2010_65 0.1313276
## SEs=
##                pop.2010
## pop_2010_65 0.002827304
```

```r
confint(svyratio(~pop_2010_65, ~pop.2010, des))
```

```
##                          2.5 %   97.5 %
## pop_2010_65/pop.2010 0.1257862 0.136869
```