Tidy Survey Analysis in R using the srvyr Package

AAPOR 2021 Short Course

Stephanie Zimmer, RTI International

Rebecca Powell, RTI International

2021-05-06

Introduction

Overview

- At the end of this course, you should be able to
    - Calculate point estimates and their standard errors with survey data
        - Means & Proportions
        - Totals
        - Quantiles
    - Perform t-tests and chi-squared tests
    - Fit regression models
    - Specify a survey design in R to create a survey object
- We will not be going over the following but provide some resources at the end
    - Weighting (calibration, post-stratification, raking, etc.)
    - Survival analysis
    - Nonlinear models

Overview: Course Roadmap

- Get familiar with RStudio Cloud with a warm-up exercise using the tidyverse
- Introduce the survey data we'll be using in the course
- Analysis of continuous data with time for practice
- Analysis of categorical data with time for practice
- Specify a survey design object in R with exercises
- Resources for other survey analysis topics
- Closing

Logistics

- We will be using RStudio Cloud today to ensure everyone has access
    - Sign-up for a free RStudio Cloud account
    - Access the project and files via link in email and Zoom chat
    - Click "START" to open the project and get started
    - Rstudio Cloud has the same features and appearance as RStudio for ease of use
- All slides and code are available on GitHub: https://github.com/szimmer/tidy-survey-aapor-2021

Intro to RStudio Cloud: Penguins!!

- Using `palmerpenguins` data for warm-up exercises
- Data were collected and made available by Dr. Kristen Gorman and the Palmer Station, Antarctica LTER, a member of the Long Term Ecological Research Network.
- Access data through `palmerpenguins` package https://github.com/allisonhorst/palmerpenguins/

If you are using your own RStudio environment:

- Make sure you have `tidyverse`, `here`, and `palmerpenguins` installed

```
# Run package installation if you don't have these packages already
# As a reminder, installing takes package from internet to your computer
# and only needs to be done once, not each session

install.packages(c("tidyverse", "here", "palmerpenguins"))
```

Intro to RStudio Cloud: Penguins!!

- Load `tidyverse`, `here`, and `palmerpenguins`
- Look at the penguins dataset using `glimpse`

```
library(tidyverse) # for tidyverse
library(here) # for file paths
library(palmerpenguins) # for warm-up data
glimpse(penguins)
```

```
## Rows: 344
## Columns: 8
## $ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adel~
## $ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgerse~
## $ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34.1, ~
## $ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18.1, ~
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, 186~
## $ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 3475, ~
## $ sex               <fct> male, female, female, NA, female, male, female, male~
## $ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007~
```

Warm-up Exercises: WarmUpExercises.Rmd

- **Let's open RStudio cloud and do some warm-up examples**
    - We will do one together and then give you 5-minutes to work through other examples and get familiar with RStudio Cloud
- Explore the penguins data
    - How many penguins of each species are there? *We will do this one together*
    - How many penguins of each species and sex are there? Hint: use `count`
    - What is the mean length of flipper by species? Hint: use `group_by` and `summarize`
    - What is the mean flipper length by species and sex?
- More advanced warm-up
    - Fit a simple linear regression between body mass and flipper length.
    - Test whether the average flipper length is significantly different between male and female penguins. Use t-test, lm, or glm

Ex. 1: How many penguins of each species are there?

```
penguins %>%
   count(species)
```

```
## # A tibble: 3 x 2
##   species       n
##   <fct>     <int>
## 1 Adelie      152
## 2 Chinstrap    68
## 3 Gentoo      124
```

Ex. 2: How many penguins of each species and sex are there?

```
penguins %>%
   count(species, sex)
```

```
## # A tibble: 8 x 3
##   species   sex        n
##   <fct>     <fct>  <int>
## 1 Adelie    female    73
## 2 Adelie    male      73
## 3 Adelie    NA         6
## 4 Chinstrap female    34
## 5 Chinstrap male      34
## 6 Gentoo    female    58
## 7 Gentoo    male      61
## 8 Gentoo    NA         5
```

Ex. 3: What is the mean length of flipper by species?

```
penguins %>%
   group_by(species) %>%
   summarize(
     MeanFlipperLength=mean(flipper_length_mm,
                            na.rm=TRUE))
```

```
## # A tibble: 3 x 2
##   species   MeanFlipperLength
##   <fct>                 <dbl>
## 1 Adelie                 190.
## 2 Chinstrap              196.
## 3 Gentoo                 217.
```

Ex. 4: What is the mean flipper length by species and sex?

```
penguins %>%
   group_by(species, sex) %>%
   summarize(
     MeanFlipperLength=mean(flipper_length_mm,
                            na.rm=TRUE))
```

```
## # A tibble: 8 x 3
## # Groups:   species [3]
##   species   sex    MeanFlipperLength
##   <fct>     <fct>              <dbl>
## 1 Adelie    female              188.
## 2 Adelie    male                192.
## 3 Adelie    NA                  186.
## 4 Chinstrap female              192.
## 5 Chinstrap male                200.
## 6 Gentoo    female              213.
## 7 Gentoo    male                222.
## 8 Gentoo    NA                  216.
```

Advanced Ex. 1: Linear regression (body mass & flipper length)

```
mod1 <- lm(body_mass_g ~ flipper_length_mm, data=penguins)
summary(mod1)
```

```
##
## Call:
## lm(formula = body_mass_g ~ flipper_length_mm, data = penguins)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1058.80  -259.27   -26.88   247.33  1288.69
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -5780.831    305.815  -18.90   <2e-16 ***
## flipper_length_mm    49.686      1.518   32.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 394.3 on 340 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.759,    Adjusted R-squared:  0.7583
## F-statistic:  1071 on 1 and 340 DF,  p-value: < 2.2e-16
```

Advanced Ex. 2: Flipper length differences by sex: t-test

```
t.test(flipper_length_mm ~ sex, data=penguins)
```

```
##
##     Welch Two Sample t-test
##
## data:  flipper_length_mm by sex
## t = -4.8079, df = 325.28, p-value = 2.336e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -10.064811  -4.219821
## sample estimates:
## mean in group female   mean in group male
##             197.3636             204.5060
```

Advanced Ex. 2: Flipper length differences by sex: lm

```
mod3 <- lm(flipper_length_mm ~ sex, data=penguins)
summary(mod3)
```

```
##
## Call:
## lm(formula = flipper_length_mm ~ sex, data = penguins)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.506 -10.364  -4.364  12.636  26.494
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  197.364      1.057 186.792  < 2e-16 ***
## sexmale        7.142      1.488   4.801 2.39e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.57 on 331 degrees of freedom
##   (11 observations deleted due to missingness)
## Multiple R-squared:  0.06511,    Adjusted R-squared:  0.06229
## F-statistic: 23.05 on 1 and 331 DF,  p-value: 2.391e-06
```

Advanced Ex. 2: Flipper length differences by sex: glm

```
mod4 <- glm(flipper_length_mm ~ sex, data=penguins)
summary(mod4)
```

```
##
## Call:
## glm(formula = flipper_length_mm ~ sex, data = penguins)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -26.506  -10.364   -4.364   12.636   26.494
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  197.364      1.057 186.792  < 2e-16 ***
## sexmale        7.142      1.488   4.801 2.39e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 184.206)
##
##     Null deviance: 65219  on 332  degrees of freedom
## Residual deviance: 60972  on 331  degrees of freedom
##   (11 observations deleted due to missingness)
## AIC: 2686
##
```

Survey Datasets

Residential Energy Consumption Survey (RECS) 2015

- Energy consumption/expenditures collected through energy suppliers
- Fielded 14 times between 1950 and 2015
- Topics include appliances, electronics, heating, a/c, temperatures, water heating, lighting, energy bills, respondent demographics, and energy assistance
- Funded by the Energy Information Administration
- **Target Population**: Primary occupied housing units in the US
- **Mode**: In-person, paper, and web interview mode
- **Sample Information**: BRR Replicate weights included for variance estimation

https://www.eia.gov/consumption/residential/index.php

American National Election Studies (ANES) 2016

- Pre and post election surveys
- Fielded almost every 2 years since 1948
- Topics include voter registration status, candidate preference, opinions on country and government, party and ideology affiliation, opinions on policy, news sources, and more
- Collaboration of Stanford, University of Michigan - funding by the National Science Foundation
- **Target Population**: US citizens, 18 and older living in US
- **Mode**: FTF with CASI and Web
- **Sample Information**: Pseudo-strata and pseudo-cluster included for variance estimation

https://electionstudies.org/

Continuous data analysis

Overview of Survey Analysis using `srvyr` Package

1. Create a `tbl_svy` object using: `as_survey_design` or `as_survey_rep`
2. Subset data (if needed) using `filter` (subpopulations)
3. Specify domains of analysis using `group_by`
4. Within `summarize`, specify variables to calculate including means, totals, proportions, quantiles and more

Note: We will be teaching this in the reverse order!!!

Set-up for Analysis

- `srvyr` package uses tidy-syntax but uses the `survey` package behind it to do calculations
- If using your own RStudio environment, install both packages:

```
# Install survey and srvyr packages

remotes::install_github("bschneidr/survey", ref = "c217689")
install.packages("srvyr")
```

- First, we will set-up a design object and later talk about what it means

```
library(survey) # for survey analysis
library(srvyr) # for tidy survey analysis

recs <- read_rds(here("Data", "recs.rds"))

recs_des <- recs %>%
   as_survey_rep(weights=NWEIGHT,
                 repweights=starts_with("BRRWT"),
                 type="Fay",
                 rho=0.5,
                 mse=TRUE)
```

Weighted Analysis for Continuous Variables

- Common functions for continuous summaries
    - survey_mean
    - survey_total (like sum)
    - survey_median
    - survey_quantile
    - survey_ratio
- Always call within `summarize`/`summarise`

`survey_mean` Syntax

```
survey_mean(
  x,
  na.rm = FALSE,
  vartype = c("se", "ci", "var", "cv"),
  level = 0.95,
  proportion = FALSE,
  deff = FALSE,
  df = NULL,
  ...
)
```

To calculate a survey mean, we use this in `summarize`/`summarise`

```
survey_design_object %>%
   summarize(
      mean_varname=survey_mean(x = continuous_varname)
      )
```

`survey_mean` Example 1: Mean dollars spent on energy

This is an example using the `recs_des` survey design object and `survey_mean` function defaults

```
recs_des %>%
   summarize(
      TD_mean=survey_mean(x = TOTALDOL)
      )
```

```
##    TD_mean TD_mean_se
## 1 1859.397   15.59328
```

`survey_mean` Example 2: Mean temperature setting for summer during the day

Run this code. What happens? Why?

```
recs_des %>%
   summarize(
      TD_mean=survey_mean(x = SummerTempDay)
      )
```

`survey_mean` Example 2: Mean temperature setting for summer during the day

Run this code. What happens? Why?

```
recs_des %>%
   summarize(
      TD_mean=survey_mean(x = SummerTempDay)
      )
```

```
## Error: Problem with `summarise()` input `TD_mean`.
## x All replicates contained NAs
## i Input `TD_mean` is `survey_mean(x = SummerTempDay)`.
```

How do we fix this code?

`survey_mean` Example 2: Missing data solution

```
recs_des %>%
   summarize(
      TD_mean = survey_mean(
        x = SummerTempDay,
        na.rm = TRUE )
      )
```

```
##    TD_mean TD_mean_se
## 1 72.42918 0.07933939
```

`survey_median` Syntax

```
survey_median(
  x,
  na.rm = FALSE,
  vartype = c("se", "ci"),
  level = 0.95,
  df = NULL,
  ...
)
```

`survey_median` Example: Median temperature setting for summer during day

Fill in the blank:

```
recs_des %>%
   summarize(
      TD_median=survey_median(x=_________,
                              na.rm=_________)
      )
```

`survey_median` Example: Median temperature setting for summer during day

Fill in the blank:

```
recs_des %>%
   summarize(
      TD_median=survey_median(x=_________,
                              na.rm=_________)
      )
```

```
recs_des %>%
   summarize(
      TD_median=survey_median(x=SummerTempDay,
                              na.rm=TRUE)
      )
```

```
##   TD_median TD_median_se
## 1        72    0.2518573
```

`survey_quantile` Syntax

```
survey_quantile(
  x,
  quantiles,
  na.rm = FALSE,
  vartype = c("se", "ci", "var", "cv"),
  level = 0.95,
  df = NULL,
  ...
)
```

`survey_quantile` Example 1: 1st and 3rd quantile of dollars spent on energy

```
recs_des %>%
   summarize(
      Spent=survey_quantile(
        x = TOTALDOL,
        quantiles = c(.25, .75))
      )
```

```
##   Spent_q25 Spent_q75 Spent_q25_se Spent_q75_se
## 1  1153.142  2353.132     13.75375     22.64873
```

`survey_quantile` Example 2: 1st and 3rd quantile of dollars spent on energy now with confidence interval

```
recs_des %>%
   summarize(
      Spent=survey_quantile(x = TOTALDOL,
                            quantiles = c(.25, .75),
                            vartype = "ci"
         )
      )
```

```
##   Spent_q25 Spent_q75 Spent_q25_low Spent_q75_low Spent_q25_upp Spent_q75_upp
## 1  1153.142  2353.132      1126.185      2308.741      1180.099      2397.523
```

`survey_ratio` Syntax

- Note this estimates: $\sum x_i / \sum y_i$ not $\sum \frac{x_i}{y_i}$

```
survey_ratio(
  numerator,
  denominator,
  na.rm = FALSE,
  vartype = c("se", "ci", "var", "cv"),
  level = 0.95,
  deff = FALSE,
  df = NULL,
  ...
)
```

`survey_ratio` Example: mean dollars per BTU spent on energy

```
recs_des %>%
   summarize(
      DolPerBTU=survey_ratio(
         numerator = TOTALDOL,
         denominator = TOTALBTU,
         na.rm = TRUE
         )
      )
```

```
##    DolPerBTU DolPerBTU_se
## 1 0.02411671 0.0002168955
```

Practice on your own

- Open ContinuousExercises.Rmd and work through Part 1
- We will take 15 minutes. Use this time for the exercises and a break

Weighted Analysis for Continuous Variables: Domain Analysis

- If we want to get estimates by another variable, we need to add a `group_by` statement before doing the analysis.
- Example: Average dollars spent on electricity by whether AC is used

```
recs_des %>%
  group_by(ACUsed) %>%
  summarize(
    ElBill=survey_mean(DOLLAREL,
                       na.rm=TRUE)
  )
```

```
## # A tibble: 2 x 3
##   ACUsed ElBill ElBill_se
##   <lgl>   <dbl>     <dbl>
## 1 FALSE    972.      25.8
## 2 TRUE    1435.      15.8
```

Domain Analysis: Totals

- If we want the overall electric bill too, use the `cascade` function instead of `summarize`

```
recs_des %>%
   group_by(ACUsed) %>%
   cascade(
      ElBill=survey_mean(DOLLAREL,
                         na.rm=TRUE)
   )
```

```
## # A tibble: 3 x 3
##   ACUsed ElBill ElBill_se
##   <lgl>   <dbl>     <dbl>
## 1 FALSE    972.      25.8
## 2 TRUE    1435.      15.8
## 3 NA      1375.      14.1
```

Domain Analysis: Totals

- Also can add sample and pop sizes

```
recs_des %>%
   group_by(ACUsed) %>%
   cascade(
      ElBill=survey_mean(DOLLAREL, na.rm=TRUE),
      N=survey_total(!is.na(DOLLAREL)),
      n=unweighted(sum(!is.na(DOLLAREL)))
   )
```

```
## # A tibble: 3 x 6
##   ACUsed ElBill ElBill_se          N        N_se     n
##   <lgl>   <dbl>     <dbl>      <dbl>       <dbl> <int>
## 1 FALSE    972.      25.8  15401242. 976901.       737
## 2 TRUE    1435.      15.8 102807008. 976901.      4949
## 3 NA      1375.      14.1 118208250.      0.0320  5686
```

Weighted Analysis for Specific Subpopulations

- filtering (subsetting) the data should be done AFTER specifying the design to ensure accurate standard errors
- Use the `filter` function after creating the survey design object and before summarizing

Wrong way:

```
data %>%
  filter(state=="NC") %>%
  as_survey_design(...) %>%
  summarize(AvgAge=mean(Age))
```

Right way:

```
data %>%
  as_survey_design(...) %>%
  filter(state=="NC") %>%
  summarize(AvgAge=mean(Age))
```

Subpopulation Example 1: Average electric cost of single family homes

```
recs_des %>%
  filter(HousingUnitType %in% c("Single-family detached",
                                "Single-family attached")) %>%
  summarize(
    ElBill=survey_mean(DOLLAREL,
                       na.rm=TRUE)
  )
```

```
##     ElBill ElBill_se
## 1 1541.933  17.24834
```

Comparisons with t-tests: `svyttest` Syntax

- t-tests are done in the package `survey` not `srvyr` but you can use the same design object

```
svyttest(formula, # outcome~group for two-sample, outcome~0 for one-sample
         design,
         na.rm = FALSE
         ....)
```

`svyttest` Example 1: One-sample t-test

- I keep my house at 68 degrees at night during the summer. Is this different from the national average?

```
recs_des %>%
   svyttest(design=.,
            formula=I(SummerTempNight-68)~0,
            na.rm=TRUE)
```

```
##
##     Design-based one-sample t-test
##
## data:  I(SummerTempNight - 68) ~ 0
## t = 41.013, df = 94, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  3.424776 3.773247
## sample estimates:
##     mean
## 3.599012
```

`svyttest` Example 2: Comparing two variables

- Do people keep their house the same temperature at night during the summer and the winter?

```
recs_des %>%
   svyttest(design=.,
            formula=I(SummerTempNight-WinterTempNight)~0,
            na.rm=TRUE)
```

```
##
##     Design-based one-sample t-test
##
## data:  I(SummerTempNight - WinterTempNight) ~ 0
## t = 29.079, df = 94, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  2.995084 3.434072
## sample estimates:
##     mean
## 3.214578
```

`svyttest` Example 3: Two-sample t-test

- Are electric bills different between those with and without A/C?

```
recs_des %>%
   svyttest(design=.,
            formula=DOLLAREL~ACUsed,
            na.rm=TRUE)
```

```
##
##     Design-based t-test
##
## data:  DOLLAREL ~ ACUsed
## t = 14.772, df = 94, p-value < 2.2e-16
## alternative hypothesis: true difference in mean is not equal to 0
## 95 percent confidence interval:
##  401.4597 524.2894
## sample estimates:
## difference in mean
##           462.8746
```

Linear Regression or ANOVA: `svyglm` Syntax

- As with t-tests, regressions are done in the package `survey` not `srvyr` but you can use the same design object
- Syntax is similar between t-test and glm

```
svyglm(formula,
       design,
       na.action, #default is na.omit
       ....)
```

`svyglm` Example: Two-sample

Same example as two-sample t-test: Are electric bills different between those with and without A/C?

t-test:

```
recs_des %>%
   svyttest(design=.,
            formula=DOLLAREL~ACUsed,
            na.rm=TRUE)
```

glm:

```
recs_des %>%
   svyglm(design=.,
          formula=DOLLAREL~ACUsed,
          na.action=na.omit)
```

`svyglm` Example: Two-sample

Are electric bills different between those with and without A/C?

```
recs_des %>%
   svyglm(design=.,
          formula=DOLLAREL~ACUsed,
          na.action=na.omit) %>%
  summary()
```

```
##
## Call:
## svyglm(design = ., formula = DOLLAREL ~ ACUsed, na.action = na.omit)
##
## Survey design:
## Called via srvyr
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   972.09      25.81   37.66   <2e-16 ***
## ACUsedTRUE    462.87      31.33   14.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 3543220488)
##
## Number of Fisher Scoring iterations: 2
```

`svyglm` Example 1: ANOVA Test

Does temperature of AC at night vary by region?

```
recs_des %>%
   svyglm(design=.,
          formula=SummerTempNight~Region,
          na.action=na.omit) %>%
  summary()
```

```
##
## Call:
## svyglm(design = ., formula = SummerTempNight ~ Region, na.action = na.omit)
##
## Survey design:
## Called via srvyr
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    70.4848     0.1968 358.151  < 2e-16 ***
## RegionMidwest   0.8744     0.2526   3.461 0.000818 ***
## RegionSouth     1.4865     0.2306   6.446 5.20e-09 ***
## RegionWest      1.6568     0.3529   4.695 9.27e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 119075)
##
## Number of Fisher Scoring iterations: 2
```

`svyglm` Example 2: Linear Model

- Is there a relationship between square footage and electric bill?
- Let's review the data first with a ggplot. *Note we use the original data and do **NOT** use the survey design object.*

```
p <- recs %>%
  ggplot(aes(x=TOTSQFT_EN, y=DOLLAREL, weight=NWEIGHT)) +
  geom_hex() +
  theme(legend.position="right") +
  guides(fill=guide_legend(title="HUs"))
```

`svyglm` Example 2: Linear Model

`svyglm` Example 2: Linear Model

```
m_electric_sqft <- recs_des %>%
   svyglm(design=.,
          formula=DOLLAREL~TOTSQFT_EN,
          na.action=na.omit)
summary(m_electric_sqft)
```

```
##
## Call:
## svyglm(design = ., formula = DOLLAREL ~ TOTSQFT_EN, na.action = na.omit)
##
## Survey design:
## Called via srvyr
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 879.89542   26.31370   33.44   <2e-16 ***
## TOTSQFT_EN    0.24633    0.01338   18.42   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 3125448288)
##
## Number of Fisher Scoring iterations: 2
```

Practice on your own

- Open ContinuousExercises.Rmd and work through Part 2
- We will take 15 minutes. Use this time for the exercises and a break

Categorical data analysis

Weighted Analysis for Categorical Variable

- Functions to use within `summarize` after `group_by`
    - survey_mean
    - survey_total
- Functions to get counts
    - survey_count

Set-up ANES Data for Examples

```
anes <- read_rds(here("Data", "anes.rds")) %>%
   mutate(Weight=Weight/sum(Weight)*224059005)
# adjust weight to sum to citizen pop, 18+ in Nov 2016 per ANES methodology documentation
anes_des <- anes %>%
   as_survey_design(weights = Weight,
                    strata = Stratum,
                    ids = VarUnit,
                    nest = TRUE)
```

`survey_count` Syntax

- `survey_count` functions similarly to `count` in that it is **NOT** called within `summarize`
- Produces weighted counts and variance of your choice of those counts

```
survey_count(
  x,
  ...,
  wt = NULL,
  sort = FALSE,
  name = "n",
  .drop = dplyr::group_by_drop_default(x),
  vartype = c("se", "ci", "var", "cv")
)
```

`survey_count` Example

- Cross-tab of population in each age group and gender

```
anes_des %>%
  survey_count(AgeGroup, Gender, name="n")
```

```
## # A tibble: 25 x 4
##    AgeGroup Gender         n     n_se
##    <fct>    <fct>      <dbl>    <dbl>
##  1 18-29    Male   23706666. 1771369.
##  2 18-29    Female 20972922. 1373206.
##  3 18-29    Other    295319.  150903.
##  4 30-39    Male   17039603. 1129924.
##  5 30-39    Female 19385497. 1249208.
##  6 30-39    Other     48425.   34492.
##  7 30-39    NA       259882.  148534.
##  8 40-49    Male   15369292. 1204485.
##  9 40-49    Female 18047731. 1205972.
## 10 40-49    Other     25736.   25736.
## # ... with 15 more rows
```

`survey_mean` and `survey_total` Examples

- `survey_mean` used with no x (variable) calculates a proportion of groups specified in `group_by`
- `survey_total` used with no x (variable) calculates a population count estimate for the groups specified in `group_by`

Cross-tab of population who voted in 2016

```
anes_des %>%
  filter(!is.na(VotedPres2016)) %>%
  group_by(VotedPres2016) %>%
  summarize(
    p=survey_mean(),
    N=survey_total(),
    n=unweighted(n()), .groups="drop"
  )
```

```
## # A tibble: 2 x 6
##   VotedPres2016     p    p_se          N     N_se     n
##   <fct>         <dbl>   <dbl>      <dbl>    <dbl> <int>
## 1 Yes           0.852 0.00768 170289743. 3798953.  2887
## 2 No            0.148 0.00768  29632875. 1669962.   444
```

Conditional proportions with more than one group

- Specifying more than one group calculates conditional proportions
- Example: people voting in 2012 and 2016

```
anes_des %>%
  filter(!is.na(VotedPres2012), !is.na(VotedPres2016)) %>%
  group_by(VotedPres2012, VotedPres2016) %>%
  summarize(
    p=survey_mean(),
    N=survey_total(),
    n=unweighted(n()), .groups="drop"
  )
```

```
## # A tibble: 4 x 7
##   VotedPres2012 VotedPres2016      p    p_se          N     N_se     n
##   <fct>         <fct>          <dbl>   <dbl>      <dbl>    <dbl> <int>
## 1 Yes           Yes           0.905  0.00717 137951845. 3318434.  2441
## 2 Yes           No            0.0948 0.00717  14455048. 1165836.   225
## 3 No            Yes           0.679  0.0199   31671027. 1780765.   435
## 4 No            No            0.321  0.0199   14980427. 1165804.   215
```

Joint proportions with more than one group

- Specify an interaction to get joint distribution
- Example: people voting in 2012 and 2016

```
anes_des %>%
  filter(!is.na(VotedPres2012), !is.na(VotedPres2016)) %>%
  group_by(groups = interaction(VotedPres2016, VotedPres2012)) %>%
  summarize(
    p=survey_mean(),
    N=survey_total(),
    .groups="drop"
  )
```

```
## # A tibble: 4 x 5
##   groups       p    p_se          N     N_se
##   <fct>    <dbl>   <dbl>      <dbl>    <dbl>
## 1 Yes.Yes 0.693  0.00971 137951845. 3318434.
## 2 No.Yes  0.0726 0.00569  14455048. 1165836.
## 3 Yes.No  0.159  0.00812  31671027. 1780765.
## 4 No.No   0.0753 0.00559  14980427. 1165804.
```

Joint proportions with more than one group

- Specify an interaction to get joint distribution
- Example: people voting in 2012 and 2016

```
anes_des %>%
  filter(!is.na(VotedPres2012), !is.na(VotedPres2016)) %>%
  group_by(groups = interaction(VotedPres2016, VotedPres2012)) %>%
  summarize(
    VotedPres2012=VotedPres2012[1],
    VotedPres2016=VotedPres2016[1],
    p=survey_mean(),
    N=survey_total(),
    .groups="drop"
  )
```

```
## # A tibble: 4 x 7
##   groups  VotedPres2012 VotedPres2016      p    p_se          N     N_se
##   <fct>   <fct>         <fct>          <dbl>   <dbl>      <dbl>    <dbl>
## 1 Yes.Yes Yes           Yes            0.693  0.00971 137951845. 3318434.
## 2 No.Yes  Yes           No             0.0726 0.00569  14455048. 1165836.
## 3 Yes.No  No            Yes            0.159  0.00812  31671027. 1780765.
## 4 No.No   No            No             0.0753 0.00559  14980427. 1165804.
```

Proportions with Design Effects

```
anes_des %>%
  filter(!is.na(VotedPres2012), !is.na(VotedPres2016)) %>%
  group_by(VotedPres2012, VotedPres2016) %>%
  summarize(
    p=survey_mean(deff=TRUE),
    N=survey_total()
  )
```

```
## # A tibble: 4 x 7
## # Groups:   VotedPres2012 [2]
##   VotedPres2012 VotedPres2016      p    p_se p_deff          N     N_se
##   <fct>         <fct>          <dbl>   <dbl>  <dbl>      <dbl>    <dbl>
## 1 Yes           Yes           0.905  0.00717   1.58 137951845. 3318434.
## 2 Yes           No            0.0948 0.00717   1.58  14455048. 1165836.
## 3 No            Yes           0.679  0.0199    1.18  31671027. 1780765.
## 4 No            No            0.321  0.0199    1.18  14980427. 1165804.
```

`svychisq` Syntax

- As with testing on continuous variables, `svychisq` comes from the `survey` package

```
svychisq(formula,
         design,
         statistic = c("F",  "Chisq", "Wald", "adjWald", "lincom", "saddlepoint"),
         na.rm=TRUE,
         ...)
```

`svychisq` Example 1: Function Defaults

- How often can you trust the federal gov't to do what is right?
- How often can you trust other people?

```
anes_des %>%
   svychisq(design=.,
            formula=~TrustPeople +TrustGovernment)
```

```
##
##     Pearson's X^2: Rao & Scott adjustment
##
## data:  NextMethod()
## F = 12.608, ndf = 13.161, ddf = 1750.368, p-value < 2.2e-16
```

`svychisq` Example 2: Wald Statistic

- How often can you trust the federal gov't to do what is right?
- Who did you vote for? Clinton, Trump, or Other

```
anes_des %>%
   svychisq(design=.,
            formula=~TrustGovernment +VotedPres2016_selection,
            statistic="Wald")
```

```
##
##     Design-based Wald test of association
##
## data:  NextMethod()
## F = 24.692, ndf = 8, ddf = 133, p-value < 2.2e-16
```

Practice on your own

- Open CategoricalExercises.Rmd and work through the exercises
- We will take 10 minutes. Use this time for the exercises and a break

Sample design object

`tbl_svy` Object: Taylor's Series

- `tbl_svy` object defines the sampling design or replicate weights
- Key information is usually found in documentation of a public use file

```
as_survey_design(
  .data,
  ids = NULL,#cluster IDs/PSUs
  strata = NULL,#strata variables
  variables = NULL,#defaults to all in .data
  fpc = NULL,#variables defining the fpc
  nest = FALSE,#TRUE/FALSE - relabel clusters to nest within strata
  check_strata = !nest, #check that clusters are nested in strata
  weights = NULL,# weight variable
  ...
)
```

`tbl_svy` for Common Designs

```
# simple random sample (SRS)
apisrs %>% as_survey_design(fpc = fpc)

# stratified sample
apistrat %>% as_survey_design(strata = stype, weights = pw)

# one-stage cluster sample
apiclus1 %>% as_survey_design(ids = dnum, weights = pw, fpc = fpc)

# two-stage cluster sample, weights computed from pop size
apiclus2 %>% as_survey_design(ids = c(dnum, snum), fpc = c(fpc1, fpc2))

# stratified, cluster design
apistrat %>% as_survey_design(ids = dnum, strata = stype, weights =pw, nest = TRUE)
```

- examples from `srvyr` help documentation

ANES Design Object

```
anes_des <- anes %>%
   as_survey_design(weights = Weight,
                    strata = Stratum,
                    ids = VarUnit,
                    nest = TRUE)
summary(anes_des)
```

```
## Stratified 1 - level Cluster Sampling design (with replacement)
## With (265) clusters.
## Called via srvyr
## Probabilities:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 2.526e-06 1.441e-05 2.176e-05       Inf 4.137e-05       Inf
## Stratum Sizes:
##             1 10 100 101 103 104 105 106 107 108 109 11 110 111 112 113 114 115
## obs        32 27  31  43  28  33  19  45  46  50  38 28  42   9  44  39  34  34
## design.PSU  2  2   2   3   2   2   2   2   2   2   2  2   2   2   2   2   2   2
## actual.PSU  2  2   2   3   2   2   2   2   2   2   2  2   2   2   2   2   2   2
##           116 117 118 119 12 120 121 122 123 124 125 126 127 128 129 13 130
## obs         42  40  52  43 36  28  55  22  57  37  44  55  32  12  41 27  14
## design.PSU   2   2   2   2  2   2   2   2   2   2   2   2   2   2   2  2   2
## actual.PSU   2   2   2   2  2   2   2   2   2   2   2   2   2   2   2  2   2
##           131 132 133 14 15 16 17 18 19  2 20 21 22 23 24 25 26 27 28 29  3 30
## obs         40  41  21 24 37 26 27 37 31 29 30 38 26 26 30 31 28 28 22 36 29 41
## design.PSU   2   2   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
## actual.PSU   2   2   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
##           31 32 33 34 35 36 37 38 39  4 40 41 42 43 44 45 46 47 48 49  5 50 51
## obs        27 30 28 28 24 37 23 32 34 23 27 36 24 38 23 30 24 25 33 32 32 24 29
```

`tbl_svy` Objects with Supplied Replicate Weights

- Key information is usually found in documentation of a public use file

```
as_survey_rep(
  .data,
  variables = NULL,#defaults to all in .data
  repweights = NULL,#variables specifying replicate weights
  weights = NULL,#variable for analysis weight
  type = c("BRR", "Fay", "JK1", "JKn", "bootstrap", "other"),
  rho = NULL,#shrinkage factor for Fay's method,
  mse = getOption("survey.replicates.mse"), #    if TRUE, compute variances based on
  # sum of squares around the point estimate, rather than the mean of the replicates
  scale = NULL, # overall multiplier for squared deviations
  ...
)
```

RECS Design Object

```
recs_des <- recs %>%
   as_survey_rep(weights=NWEIGHT,
                 repweights=starts_with("BRRWT"),
                 type="Fay",
                 rho=0.5,
                 mse=TRUE)
summary(recs_des)
```

```
## Call: Called via srvyr
## Fay's variance method (rho= 0.5 ) with 96 replicates and MSE variances.
## Sampling variables:
##  - repweights: `BRRWT1 + BRRWT2 + BRRWT3 + BRRWT4 + BRRWT5 + BRRWT6 + BRRWT7 + BRRWT8 + BRRWT9 + BRRWT10 + BRRWT11 + BRRWT12 + BRRWT13
##  - weights: NWEIGHT
## Data variables: DOEID (dbl), Region (fct), Division (fct), MSAStatus (fct),
##   Urbanicity (fct), HousingUnitType (fct), YearMade (ord), SpaceHeatingUsed
##   (lgl), HeatingBehavior (fct), WinterTempDay (dbl), WinterTempAway (dbl),
##   WinterTempNight (dbl), ACUsed (lgl), ACBehavior (fct), SummerTempDay (dbl),
##   SummerTempAway (dbl), SummerTempNight (dbl), TOTCSQFT (dbl), TOTHSQFT (dbl),
##   TOTSQFT_EN (dbl), TOTUCSQFT (dbl), TOTUSQFT (dbl), NWEIGHT (dbl), BRRWT1
##   (dbl), BRRWT2 (dbl), BRRWT3 (dbl), BRRWT4 (dbl), BRRWT5 (dbl), BRRWT6 (dbl),
##   BRRWT7 (dbl), BRRWT8 (dbl), BRRWT9 (dbl), BRRWT10 (dbl), BRRWT11 (dbl),
##   BRRWT12 (dbl), BRRWT13 (dbl), BRRWT14 (dbl), BRRWT15 (dbl), BRRWT16 (dbl),
##   BRRWT17 (dbl), BRRWT18 (dbl), BRRWT19 (dbl), BRRWT20 (dbl), BRRWT21 (dbl),
##   BRRWT22 (dbl), BRRWT23 (dbl), BRRWT24 (dbl), BRRWT25 (dbl), BRRWT26 (dbl),
##   BRRWT27 (dbl), BRRWT28 (dbl), BRRWT29 (dbl), BRRWT30 (dbl), BRRWT31 (dbl),
##   BRRWT32 (dbl), BRRWT33 (dbl), BRRWT34 (dbl), BRRWT35 (dbl), BRRWT36 (dbl),
##   BRRWT37 (dbl), BRRWT38 (dbl), BRRWT39 (dbl), BRRWT40 (dbl), BRRWT41 (dbl),
##   BRRWT42 (dbl), BRRWT43 (dbl), BRRWT44 (dbl), BRRWT45 (dbl), BRRWT46 (dbl),
```

Create Replicate Weights: jackknife

- You can also start with a design object specified by the design and create replicate weights

```
data(api)
dclus1 <- apiclus1 %>% as_survey_design(ids = dnum, weights = pw, fpc = fpc)
rclus1 <- as_survey_rep(dclus1)
summary(rclus1)
```

```
## Call: Called via srvyr
## Unstratified cluster jacknife (JK1) with 15 replicates.
## Data variables: cds (chr), stype (fct), name (chr), sname (chr), snum (dbl),
##   dname (chr), dnum (int), cname (chr), cnum (int), flag (int), pcttest (int),
##   api00 (int), api99 (int), target (int), growth (int), sch.wide (fct),
##   comp.imp (fct), both (fct), awards (fct), meals (int), ell (int), yr.rnd
##   (fct), mobility (int), acs.k3 (int), acs.46 (int), acs.core (int), pct.resp
##   (int), not.hsg (int), hsg (int), some.col (int), col.grad (int), grad.sch
##   (int), avg.ed (dbl), full (int), emer (int), enroll (int), api.stu (int), fpc
##   (dbl), pw (dbl)
## Variables:
##  [1] "cds"      "stype"    "name"     "sname"    "snum"     "dname"
##  [7] "dnum"     "cname"    "cnum"     "flag"     "pcttest"  "api00"
## [13] "api99"    "target"   "growth"   "sch.wide" "comp.imp" "both"
## [19] "awards"   "meals"    "ell"      "yr.rnd"   "mobility" "acs.k3"
## [25] "acs.46"   "acs.core" "pct.resp" "not.hsg"  "hsg"      "some.col"
## [31] "col.grad" "grad.sch" "avg.ed"   "full"     "emer"     "enroll"
## [37] "api.stu"  "fpc"      "pw"
```

Create Replicate Weights: bootstrap

- You can also start with a design object specified by the design and create replicate weights

```
bclus1 <- as_survey_rep(dclus1, type="bootstrap", replicates=100)
summary(bclus1)
```

```
## Call: Called via srvyr
## Survey bootstrap with 100 replicates.
## Data variables: cds (chr), stype (fct), name (chr), sname (chr), snum (dbl),
##   dname (chr), dnum (int), cname (chr), cnum (int), flag (int), pcttest (int),
##   api00 (int), api99 (int), target (int), growth (int), sch.wide (fct),
##   comp.imp (fct), both (fct), awards (fct), meals (int), ell (int), yr.rnd
##   (fct), mobility (int), acs.k3 (int), acs.46 (int), acs.core (int), pct.resp
##   (int), not.hsg (int), hsg (int), some.col (int), col.grad (int), grad.sch
##   (int), avg.ed (dbl), full (int), emer (int), enroll (int), api.stu (int), fpc
##   (dbl), pw (dbl)
## Variables:
##  [1] "cds"      "stype"    "name"     "sname"    "snum"     "dname"
##  [7] "dnum"     "cname"    "cnum"     "flag"     "pcttest"  "api00"
## [13] "api99"    "target"   "growth"   "sch.wide" "comp.imp" "both"
## [19] "awards"   "meals"    "ell"      "yr.rnd"   "mobility" "acs.k3"
## [25] "acs.46"   "acs.core" "pct.resp" "not.hsg"  "hsg"      "some.col"
## [31] "col.grad" "grad.sch" "avg.ed"   "full"     "emer"     "enroll"
## [37] "api.stu"  "fpc"      "pw"
```

Create Survey Design Object for ACS

Fill in the blanks

- Analysis weight: PWGTP
- replicate weights: PWGTP1-PWGTP180
- jackknife with scale adjustment of 4/80

```
acs_des <- acs_pums %>%
  as_survey_rep(
    weights=___________,
    repweights=___________,
    type=___________,
    scale=_________
  )
```

Create Survey Design Object for ACS

Fill in the blanks

- Analysis weight: PWGTP
- replicate weights: PWGTP1-PWGTP180
- jackknife with scale adjustment of 4/80

```
acs_des <- acs_pums %>%
  as_survey_rep(
    weights=___________,
    repweights=___________,
    type=___________,
    scale=_________
  )
```

```
acs_des <- acs_pums %>%
   as_survey_rep(
      weights=PWGTP,
      repweights=stringr::str_c("PWGTP", 1:80),
      type="JK1",
      scale=4/80
   )
```

Create Survey Design Object for CPS 2011 Supplement

Fill in the blanks

- Analysis weight: wtsupp
- replicate weights: repwtp1 -repwtp160
- BRR

```
cps_des <- cps %>%
  as_survey_rep(
    weights=___________,
    repweights=___________,
    type=___________
  )
```

Create Survey Design Object for CPS 2011 Supplement

Fill in the blanks

- Analysis weight: wtsupp
- replicate weights: repwtp1 -repwtp160
- BRR

```
cps_des <- cps %>%
  as_survey_rep(
    weights=___________,
    repweights=___________,
    type=___________
  )
```

```
cps_des <- cps %>%
  as_survey_rep(
    weights=wtsupp,
    repweights=starts_with("repwtp"),
    type="BRR"
  )
```

Create Survey Design Object for NHANES

Fill in the blanks

- Analysis weight: WTINT2YR
- Variance Stratum: SDMVSTRA
- Variance Primary Sampling Unit: VPSU

```
nhanes_des <- nhanes %>%
  as_survey_design(
    weights=___________,
    ids=___________,
    strata=___________,
    fpc=___________
  )
```

Create Survey Design Object for NHANES

Fill in the blanks

- Analysis weight: WTINT2YR
- Variance Stratum: SDMVSTRA
- Variance Primary Sampling Unit: VPSU

```
nhanes_des <- nhanes %>%
  as_survey_design(
    weights=___________,
    ids=___________,
    strata=___________,
    fpc=___________
  )
```

```
nhanes_des <- nhanes %>%
  as_survey_design(
    weights=WTINT2YR,
    ids=VPSU,
    strata=SDMVSTRA,
    fpc=NULL
  )
```

Create Survey Design Object for LEMAS 2016

Fill in the blanks

- Analysis weight: ANALYSISWEIGHT
- Variance Stratum: STRATA
- FPC: FRAMESIZE

```
lemas_des <- lemas %>%
  as_survey_design(
    weights=___________,
    ids=___________,
    strata=___________,
    fpc=___________
  )
```

Create Survey Design Object for LEMAS 2016

Fill in the blanks

- Analysis weight: ANALYSISWEIGHT
- Variance Stratum: STRATA
- FPC: FRAMESIZE

```
lemas_des <- lemas %>%
  as_survey_design(
    weights=___________,
    ids=___________,
    strata=___________,
    fpc=___________
  )
```

```
lemas_des <- lemas %>%
  as_survey_design(
    weights=ANALYSISWEIGHT,
    ids=1,
    strata=STRATA,
    fpc=FRAMESIZE
  )
```

Closing

Resources for more learning

- https://cran.r-project.org/web/packages/srvyr/vignettes/srvyr-vs-survey.html
- https://r-survey.r-forge.r-project.org/survey/
    - Includes more advanced modeling

Thank You!

We hope you learned a lot in this short course!

Please let us know if you have any feedback on this course. You will receive an email from AAPOR asking you to fill out a survey about this course. All feedback is welcome!

Questions?

Sources

- The American National Election Studies (https://electionstudies.org/). These materials are based on work supported by the National Science Foundation under grant numbers SES 1444721, 2014-2017, the University of Michigan, and Stanford University.
- *Residential Energy Consumption Survey: Using the 2015 Microdata File to Compute Estimates and Standard Errors.* U.S. Department of Energy (2017) https://www.eia.gov/consumption/residential/data/2015/pdf/microdata_v3.pdf
- Horst AM, Hill AP, Gorman KB (2020). palmerpenguins: Palmer Archipelago (Antarctica) penguin data. R package version 0.1.0. https://allisonhorst.github.io/palmerpenguins/
- T. Lumley (2020) "survey: analysis of complex survey samples". R package version 4.0. https://r-survey.r-forge.r-project.org/survey/
- Greg Freedman Ellis and Ben Schneider (2020). srvyr: 'dplyr'-Like Syntax for Summary Statistics of Survey Data. R package version 1.0.0. https://CRAN.R-project.org/package=srvyr
- Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2021). dplyr: A Grammar of Data Manipulation. R package version 1.0.5. https://CRAN.R-project.org/package=dplyr