Tidy Survey Analysis in R using the srvyr Package

Workshop Day 2 - Continuous Data

Stephanie Zimmer, Abt Associates

Rebecca Powell, RTI International

Isabella Velásquez, RStudio

April 22, 2022

Introduction

Overview

- At the end of this workshop series, you should be able to
    - Calculate point estimates and their standard errors with survey data
        - Proportions, totals, and counts
        - Means, quantiles, and ratios
    - 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

About Us

Stephanie Zimmer
Abt Associates
Twitter: @statsteph

Rebecca Powell
RTI International
RStudio Education Profile

Isabella Velásquez
RStudio
Twitter: @ivelasq3

About Us

Stephanie Zimmer
Abt Associates
Twitter: @statsteph

Rebecca Powell
RTI International
RStudio Education Profile

Isabella Velásquez
RStudio
Twitter: @ivelasq3

Thank you to our volunteers!

Greg Freedman-Ellis, **Raphael Nishimura**, and **Ben Schneider** will be assisting during our breakout rooms.

About This Workshop

- Hosted by Midwest Association for Public Opinion Research (MAPOR), a regional chapter of the American Association for Public Opinion Research (AAPOR).
- Originally delivered at AAPOR Conference in May 2021

Upcoming Work

- Book on analyzing survey data in R, published by CRC, Taylor & Francis Group
- We would love your help! After each course, we will send out a survey to gather your feedback on the material, organization, etc.
- Keep updated by following our project on GitHub: https://github.com/tidy-survey-r

Overview

- Last week we introduced how to do survey analysis in R with categorical data
- Today we focus on continuous data
- At the end of today, you should be able to
    - Calculate means and quantiles with their standard errors
    - Perform t-tests
    - Fit linear regression models
- Next week, we will discuss
    - Specifying a survey design object
    - Creating replicate weights
    - Creating derived/analysis/recoded variables
    - Reproducibility

Overview: Workshop 2 Roadmap

- Quick refresh of RStudio Cloud with a warm-up exercise using the tidyverse
- Introduce the survey data we'll be using today
- Calculate point estimates for continuous data with time for practice
- Significance testing with t-test and linear regression models with time for practice
- Closing

Logistics

- We will be using RStudio Cloud today to ensure everyone has access
- Sign-up for a free RStudio Cloud account (https://rstudio.cloud/)
- 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/tidy-survey-r/tidy-survey-short-course

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, Adelie, Adelie, Adelie, Adeli
## $ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgersen, Torgersen, Torgersen,
## $ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34.1, 42.0, 37.8, 37.8, 41.1, 3
## $ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18.1, 20.2, 17.1, 17.3, 17.6, 2
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, 186, 180, 182, 191, 198, 185
## $ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 3475, 4250, 3300, 3700, 3200, 3
## $ sex               <fct> male, female, female, NA, female, male, female, male, NA, NA, NA, NA, female,
## $ year              <int> 2007, 2007, 2007, 2007, 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**
- Take 8 minutes to do these exercises in breakout rooms.
- Explore the penguins data
  - What is the mean body mass in grams of all penguins? Hint: use `summarize` and remove missing data
  - What is the mean length of flipper by species? Hint: use `group_by`
  - What is the mean flipper length by species and sex?
  - Fit a simple linear regression between body mass and flipper length.

Ex. 1: What is the mean body mass in grams of all penguins? Hint: use **`summarize`** and remove missing data

```
penguins %>%
   summarize(
     MeanBodyMass=mean(body_mass_g,
                       na.rm=TRUE))
```

```
## # A tibble: 1 x 1
##   MeanBodyMass
##          <dbl>
## 1        4202.
```

Ex. 2: What is the mean length of flipper by species? Hint: use **`group_by`**

```
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. 3: 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.
```

Ex. 4: Fit a simple linear regression between body mass and 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
```

Continuous survey 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, medians, quantiles and more

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

Dataset: 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

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/r-forge-survey-mirror")
install.packages("srvyr")
```

- First, we will set-up a design object and talk about what it means in Session 3

```
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: On average, how much do US households spend on energy each year?

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)
      )
```

```
## # A tibble: 1 x 2
##   TD_mean TD_mean_se
##     <dbl>      <dbl>
## 1   1859.       15.6
```

`survey_mean` Example 2: What is the average temperature US households set their homes to on a summer day?

Run this code. What happens?

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

`survey_mean` Example 2: What is the average temperature US households set their homes to on a summer day?

Run this code. What happens?

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

```
## Error in `dplyr::summarise()`:
## ! Problem while computing `TD_mean = survey_mean(x = SummerTempDay)`.
## Caused by error in `svrVar()`:
## ! All replicates contained NAs
```

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 )
      )
```

```
## # A tibble: 1 x 2
##   TD_mean TD_mean_se
##     <dbl>      <dbl>
## 1    72.4     0.0793
```

`survey_median` Syntax

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

`survey_median` Example: What is the median temperature US households set their homes to on a summer day?

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

`survey_median` Example: What is the median temperature US households set their homes to on a summer day?

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

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

```
## # A tibble: 1 x 2
##   TD_median TD_median_se
##       <dbl>        <dbl>
## 1        72        0.252
```

`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: What are the 1st and 3rd quantile of dollars spent on energy?

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

```
## # A tibble: 1 x 4
##   Spent_q25 Spent_q75 Spent_q25_se Spent_q75_se
##       <dbl>     <dbl>        <dbl>        <dbl>
## 1     1153.     2353.         13.9         22.7
```

`survey_quantile` Example 2: What are the 1st and 3rd quantile of dollars spent on energy with confidence intervals?

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

```
## # A tibble: 1 x 6
##   Spent_q25 Spent_q75 Spent_q25_low Spent_q75_low Spent_q25_upp Spent_q75_upp
##       <dbl>     <dbl>         <dbl>         <dbl>         <dbl>         <dbl>
## 1     1153.     2353.         1124.         2310.         1179.         2400.
```

`survey_quantile` Updated Output

```
recs_des %>%
   summarize(
      Spent=survey_quantile(x = TOTALDOL,
                            quantiles = c(.25, .75),
                            vartype = "ci"
         )
      ) %>%
   pivot_longer(cols=c(Spent_q25:Spent_q75_upp),
                names_to="varname",values_to="value") %>%
   mutate(Quantile=paste0(str_sub(varname,8,9),"th Quantile"),
          type=case_when(str_detect(varname,"_low")~"Lower_Bound",
                         str_detect(varname,"_upp")~"Upper_Bound",
                         TRUE~"Estimate")) %>%
   pivot_wider(id_cols=Quantile,names_from=type,values_from=value)
```

```
## # A tibble: 2 x 4
##   Quantile      Estimate Lower_Bound Upper_Bound
##   <chr>            <dbl>       <dbl>       <dbl>
## 1 25th Quantile    1153.       1124.       1179.
## 2 75th Quantile    2353.       2310.       2400.
```

`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: What is the average dollar per BTU spent on energy?

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

```
## # A tibble: 1 x 2
##   DolPerBTU DolPerBTU_se
##       <dbl>        <dbl>
## 1    0.0241     0.000217
```

Breakout rooms: Practice time

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

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: What is the average amount of dollars spent on electricity for households that use AC and those that do not use AC?

```
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 average 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
```

Note: The overall average electric bill appears as NA

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: 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)
  )
```

```
## # A tibble: 1 x 2
##   ElBill ElBill_se
##    <dbl>     <dbl>
## 1  1542.      17.2
```

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` Syntax with %>%

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

`svyttest` Syntax with %>%

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

`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:
##  400.6588 525.0903
## 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(color="white") +
  scale_fill_gradient(guide="colourbar",name="Count of Housing Units",labels=comma)
```

`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
```

Breakout rooms: Practice time

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

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 session!

Please let us know if you have any feedback on this workshop. All feedback is welcome!

Questions?

Sources

- *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

Session info - platform

```
##  setting  value
##  version  R version 4.1.3 (2022-03-10)
##  os       Windows 10 x64 (build 19042)
##  system   x86_64, mingw32
##  ui       RStudio
##  language (EN)
##  collate  English_United States.1252
##  ctype    English_United States.1252
##  tz       America/New_York
##  date     2022-04-14
##  rstudio  2022.02.1+461 Prairie Trillium (desktop)
##  pandoc   2.17.1.1 @ C:/Program Files/RStudio/bin/quarto/bin/ (via rmarkdown)
```

Session info - packages

```
##  package         * version date (UTC) lib source
##  dplyr           * 1.0.8   2022-02-08 [1] CRAN (R 4.1.2)
##  DT              * 0.21    2022-02-26 [1] CRAN (R 4.1.3)
##  forcats         * 0.5.1   2021-01-27 [1] CRAN (R 4.1.2)
##  ggplot2         * 3.3.5   2021-06-25 [1] CRAN (R 4.1.2)
##  here            * 1.0.1   2020-12-13 [1] CRAN (R 4.1.2)
##  hexbin          * 1.28.2  2021-01-08 [1] CRAN (R 4.1.2)
##  knitr           * 1.37    2021-12-16 [1] CRAN (R 4.1.2)
##  Matrix          * 1.4-0   2021-12-08 [2] CRAN (R 4.1.3)
##  palmerpenguins * 0.1.0   2020-07-23 [1] CRAN (R 4.1.2)
##  purrr           * 0.3.4   2020-04-17 [1] CRAN (R 4.1.2)
##  readr           * 2.1.2   2022-01-30 [1] CRAN (R 4.1.2)
##  remotes         * 2.4.2   2021-11-30 [1] CRAN (R 4.1.2)
##  scales          * 1.1.1   2020-05-11 [1] CRAN (R 4.1.2)
##  srvyr           * 1.1.1   2022-02-20 [1] CRAN (R 4.1.3)
##  stringr         * 1.4.0   2019-02-10 [1] CRAN (R 4.1.2)
##  survey          * 4.2     2022-03-31 [1] Github (bschneidr/r-forge-survey-mirror@69c62ff)
##  survival        * 3.2-13  2021-08-24 [2] CRAN (R 4.1.3)
##  tibble          * 3.1.6   2021-11-07 [1] CRAN (R 4.1.2)
##  tidyr           * 1.2.0   2022-02-01 [1] CRAN (R 4.1.2)
##  tidyverse       * 1.3.1   2021-04-15 [1] CRAN (R 4.1.2)
##  xaringan        * 0.23    2022-03-08 [1] CRAN (R 4.1.3)
##
```