Analyzing Survey Data

Joseph Nathan Cohen

11/3/2019

Contents

Introduction

This week, we begin our first advanced topic: complex survey data analysis. As with all of these topics, my goal is to give you an introduction and basic exposure to this topic. This introduction is intended to orient you towards further research and practice in the topic, should your work deem these skills necessary. There is much more to learn, however.

For further study, I recommend this book: Thomas Lundley (2011) *Complex Surveys: A Guide to Analysis Using R* Wiley. ISBN 978-0470284308

We will begin our lesson by discussing the importance of sampling method, representation, and weighting in survey data analysis. Then we will look at how to execute and interpret survey data in R, and walk through an extended example using data from the General Social Survey.

Sampling, Representation, and Weights

In research, we seek to make inferences or characterizations of a *population*, or a group of people in society. To make these inferences, we try to draw a *sample*, or a subset of members from the larger population. These samples give us a sense of what kinds of people are part of the population, and how they think or behave.

In your research methods class, you learned that you want your sample to be selected randomly. A sample is *random* when every member of the population about which you wish to generalize has an equal change of being selected to be observed. If you wish to generalize about Queens College students, then your sampling mechanism should give every QC student an equal change of being asked. If you wish to generalize about Americans, every American needs an equal change to be asked.

A random sample may be more feasble were you to survey QC students (get a list of students from the Registrar, number them, and ask whoever's number comes up in a random number generator). However, the task is considerably more difficult when trying to sample from the entire city of New York, or even all the United States. How do you make people without phone numbers or addresses eligible to answer your questions? How can a surveyor intercept someone who systematically avoids human interaction? The General Social Survey can't do it on a budget of several thousand dollars per respondent.

To correct for this fundamental weakness in sampling, survey analysts employ weights. *Survey weights* are calculations used to correct for the over- or under-representation of groups in our samples. A very simple example: Imagine that the only difference between people is whether they are old or young. Say that we are surveying a population that is 60% young and 40% old. Now imagine we put out a survey using a sampling mechanism that renders 50%-50% split between old and young. In this circumstance, we would count each young person as contributing the equivalent of 1.2 hypothetical pseudo-people's worth of information:

$$\frac{Population.Proportion}{Sample.Proportion} = \frac{60}{50} = 1.2$$

Likewise, we would could each old respondent as $40/50 = 0.8$ people. The result is that our statistical findings will reflect subjects' responses in proportion with their overall representation in the population we are studying.

We can extend this process to include more categories of people. Perhaps we divide this population into another dichotomous group (say, male and female) and then adjust for the proper proportion of young males, old males, young females, and old females. We can add education level, race, and any other number of variables for which we have collected data and know the true population proportions.

In this lesson, we are going to concentrate on working with data for which weights have already been created. We will bracket the issue of creating weights. For more, see Lundley (2011) and Risto Lehtonen and Erkki Pahkinen (2004) *Practical Methods for Design and Analysis of Complex Surveys* Wiley. Today, my focus is on how to analyze survey data for which other analysts have already calculated weights.

Online Compendium of Scripts for Analyzing Major Survey Data

There is an excellent resource available online courtesy of data analyst Anthony Joseph Damico, an analyst at the Kaiser Foundation. He has created an excellent web site with instructions on analyzing many of the major workhorse surveys in social science research. Visit asdfree.com, where you will find highly useful scripts.

Data

We are going to revisit last week's data from the General Social Survey, which I downloaded from Berkeley SDA (https://sda.berkeley.edu/). I am giving you the raw data and codebook. Here is are the codes that I used to clean the data:

```
DAT <- read.csv("GSS Extract 8.csv")
DAT <- DAT[-c(1,17,36:60)]

DAT$AGE <- ifelse(DAT$AGE > 97, NA, DAT$AGE)
DAT$SEX <- factor(DAT$SEX, labels = c("Male", "Female"))
DAT$RACE <- ifelse(DAT$RACE == 0, NA, DAT$RACE)
DAT$RACE <- factor(DAT$RACE, labels = c("White", "Black", "Other"))
temp <- c("DEGREE", "PADEG", "MADEG", "SPDEG")
DAT[temp] <- lapply(DAT[temp], function(x) ifelse(x %in% c(7:9), 0, x))
DAT[temp] <- lapply(DAT[temp], function(x) factor(x,
                                                  labels = c("Below High School",
                                                             "High School",
                                                             "Junior College",
                                                             "Bachelors", "Graduate"),
                                                  ordered = T))
DAT$DWELLING <- ifelse(DAT$DWELLING %in% c(0, 98, 99), NA, DAT$DWELLING)
DAT$WORDSUM <- ifelse(DAT$WORDSUM %in% c(-1, 99), NA, DAT$WORDSUM)
DAT$WRKSTAT <- ifelse(DAT$WRKSTAT %in% c(0,9), NA, DAT$WRKSTAT)
DAT$WRKSTAT <- factor(DAT$WRKSTAT, labels = c("Working Full-Time", "Working Part-Time",
                                              "Temp Not Working","Unemployed", "Retired",
                                              "School", "Keeping House", "Other"))
DAT$MARITAL <- ifelse(DAT$MARITAL == 9, NA, DAT$MARITAL)
DAT$MARITAL <- factor(DAT$MARITAL, labels = c("Married", "Widowed", "Divorced",
                                              "Separated", "Never Married"))
DAT$CHILDS <- ifelse(DAT$CHILDS == 9, NA, DAT$CHILDS)
DAT$AGEKDBRN <- ifelse(DAT$AGEKDBRN > 97, NA, DAT$AGEKDBRN)
DAT$REGION <- ifelse(DAT$REGION == 0, NA, DAT$REGION)
DAT$REGION <- factor(DAT$REGION, labels = c("New England", "Mid Atlantic", "EN Central",
                                            "WN Central","South Atlantic", "ES Central",
                                            "WS Central", "Mountain", "Pacific"))
DAT$SIZE <- ifelse(DAT$SIZE <= 0, NA, DAT$SIZE)
DAT$PARTYID <- ifelse(DAT$PARTYID %in% c(7:9), NA, DAT$PARTYID)
DAT$PARTYID <- factor(DAT$PARTYID, labels = c("Strong Dem", "Mod Dem", "Leans Dem",
                                              "Indep.", "Leans Rep", "Mod Rep",
                                              "Strong"))
temp <- names(DAT)[19:33]
DAT[temp] <- lapply(DAT[temp], function(x) ifelse(x %in% c(0, 8, 9), NA, x))
DAT[temp] <- lapply(DAT[temp], function(x) factor(x,  labels = c("Too Little",
                                                                 "About Right",
                                                                 "Too Much")))
temp <- names(DAT)[34:46]
DAT[temp] <- lapply(DAT[temp], function(x) ifelse(x %in% c(0, 8, 9), NA, x))
DAT[temp] <- lapply(DAT[temp], function(x) factor(x, labels = c("A Great Deal",
                                                                "Only Some",
                                                                "Hardly Any")))
temp <- names(DAT)[51:56]
DAT[temp] <- lapply(DAT[temp], function(x) ifelse(x <= 0, NA, x))
```

Preparing Your Analysis

A professionally-produced data set will have proper documentation, which includes information on the sampling scheme, response rates, and weighting schemes employed its production. The absence of this information should be treated as a red flag, suggesting a possibility of low quality data. (In general, I am not interested in analyzing undocumented secondary data. There's no way to be sure it's not total garbage.) This lesson's data set comes from the National Opinion Research Center at the University of Chicago. It is among the polling industry's most respected shops, and its data is well documented.

If you are going to run an analytical project using well-documented data (like in your thesis), you have to read the documentation through. Our lesson's documentation is available online at gss.norc.org/get-documentation). Information on sampling and weighting is given in Appendix A. This document covers all data since the 1970s, and describes how sampling schemes have evolved over forty-ish years. Today's analysis uses 2016 data, which employs the 2010 National sample Design.

When judging a data set, you should marshall what you learned of proper sampling, sample bias, etc. from your research methodology classes. If you still feel shaky on this material, any goods methods text can give you a refresher. Before delving into the numbers, ask yourself if you see critical sources of bias in the survey's questions, sampling method, or response acquisition methods. Make a critical judgment as to whether the data is meritorious or garbage. To the extent that it is garbage, you should ask yourself if it is worth analyzing at all. You know what they say: *garbage in, garbage out*. This probably holds no matter how skilled the statistician analyzing the data may be.

For the purposes of processing the numbers, look for these pieces of information when reading the documentation:

- ***Sampling Scheme.*** How potential respondents were approached. Was the sample *stratified*[1]? Did the scheme involve a *clustering* strategy[2] Or was the scheme a simple random selection through some process that does not group people[3]? Check to see if there is *oversampling*, where a survey targets a smaller but theoretically-important sub-group.
- ***Primary Sampling Unit (PSU).*** In multistage sampling schemes, the PSU is the highest-order cluster in the data, from which lower-level clusters are selected. For example, in a sampling scheme that chooses states, then school districts, then schools, then classes, then students, the PSU is *states*. A survey that randomly selects zip codes, then blocks, then households, and then household members has *zip codes* as its PSU.
- ***Nonresponse Rates.*** How often were selected respondents unwilling to complete the survey? If this number is too low, it is possible that only highly-motivated respondent answered the question, which may in turn produce unseen bias.[4] The GSS is a gold standard survey, generally registers a response rate of over 70%. Many industry surveys have response rates below 10% or even 5%.
- ***Weighting Variables.*** The documentation should include a clear explanation of what the data set's weighting variable scores mean, and how they were calculated. For our set, these issues are discussed on page 3186 of the documentation, under *"Weights for 2004+ GSS"*

[1] First divided into larger, mutually-exclusive and mutually-exclusive groups from which lower order units (e.g., clusters or respondents) are chosen.

[2] Whereby the population is divided into natural groups (e.g., households, schools), from which lower-order units (e.g., household members, students) are chosen?

[3] For example, people chosen by a random number generator off of a single sampling frame.

[4] Remember that weighting only mitigates for poor representation across measured variables. A very low response rate might mean that only people leading certain lifestyles (e.g., not working when the surveyor calls or visits), certain personality proclivities (e.g., their penchant answering unknown phone numbers, giving information to strangers), or some other unmeasured characteristic is answering your data's questions. The bias is akin to those found on online rating sites like Yelp or Amazon, where most ratings are either highly positive or negative, mainly because people with middling opinions aren't motivated to leave ratings.

Through your reading, you should at least find a weight variable attached to individual respondents. You may also find variables that encode the stratum, primary sampling unit, and other cluster(s) to which each observation pertains. In this case, a read of the documentation identifies:

- A unit-level weighting variable (WTSSALL).
- A variable demarcating the strata used in this survey (VSTRAT), and
- A cluster identifier (VPSU)

Once we understand the structure of the sampling mechanism and the variables that encode this scheme, we are ready to move to the next step.

Create the Survey Design Object

R has specialty packages that facilitate survey analysis. Today, we will use the ***survey*** package.

```
library(survey)
```

Your first step is to create a "survey design object", which our survey analysis library uses to store both data and information on its weights, strata, and clusters. It then applies all this information to our statistical calculations in order to ensure that our analysis yields proper population estimates. Note the tilde (~) before variable names.

The command is:

```
OBJECT.NAME <- svydesign(
  ids = ~[CLUSTER IDs, FROM LARGEST TO SMALLEST],
  strata = ~[STRATUM VARIABLE],
  data = [DATA OBJECT],
  weights = ~[WEIGHTS VARIBLE],
  nest = {TRUE IF YOU WANT TO USE CLUSTER IDs TO NEST WITHIN STRATA}
)
```

So, for our data:

```
gss.design <- svydesign(
  ids = ~VPSU,
  strata = ~VSTRAT,
  data = DAT,
  weights = ~WTSSALL,
  nest = T)
```

To get information on the object:

```
summary(gss.design)
```

Creating Subsets of Observations

If you want to restrict your analysis to a particular sub-group, use the **subset()** command:

```
gss.design.young <- subset(gss.design, AGE < 30)
```

This creates a "Young" design object that only includes respondents under age 30:

```
svymean(~AGE, gss.design, na.rm = T)
```

```
##       mean     SE
## AGE 47.561 0.4037
```

```
svymean(~AGE, gss.design.young, na.rm = T)
```

```
##       mean     SE
## AGE 24.012 0.1968
```

If we adjust our calculations for survey representation, our data suggest an average age of 47.6 years in our overall sample, and 24.0 years in our under-30 sample.

Creating & Recoding Data

If you wish create or recode variables, you use the **update()** function on the survey design object.

Imagine we wished to compare the proportion of seniors to non-seniors using this data. If you use the unweighted data, you are told that 21.6% of the population are seniors:

```
seniors <- ifelse(DAT$AGE >= 65, 1, 0)
prop.table(table(seniors))
```

```
## seniors
##         0         1
## 0.7840392 0.2159608
```

I can calculate that new variable within the survey design object, and then calculate estimates that are reweighted:

```
gss.design.2 <- update(gss.design,
                       SENIORS = ifelse(AGE >= 65, 1, 0))
gss.design.2 <- update(gss.design.2,
                       SENIORS = factor(SENIORS))
svymean(~SENIORS, gss.design.2, na.rm=T)
```

```
##              mean     SE
## SENIORS0 0.81436 0.0088
## SENIORS1 0.18564 0.0088
```

Using a method that recalibrates for observed over-/under-representation, we get an estimate that 18.6% are seniors. This should be closer to other high-quality estimates of the older population (e.g., those from the Census Bureau).

Descriptive Statistics

Population Proportion Estimates

Frequency tables are a good way to generate numerical descriptors of a single discrete variable. Make sure that your variable is specified as a factor. We use the **svymean()** command to get the population's expected proportions and these estimates' standard errors:

```
tab.1 <- svymean(~ DEGREE, gss.design, na.rm = T)
tab.1
```

```
##                              mean     SE
## DEGREEBelow High School 0.120973 0.0096
## DEGREEHigh School       0.516686 0.0129
## DEGREEJunior College    0.074508 0.0056
## DEGREEBachelors         0.183143 0.0101
## DEGREEGraduate          0.104690 0.0077
```

Population Mean Estimates

Again, we use **svymean()**. It will recognize whether or not we are dealing with a factor or numerical variable:

```
svymean(~ AGE, gss.design, na.rm = T)
```

```
##       mean     SE
## AGE 47.561 0.4037
```

Population Percentile Score Estimates

We use **svyquantile()**. To obtain the 25th percentile score of the variable WORDSUM:

```
svyquantile( ~ WORDSUM , gss.design , 0.25 , na.rm = TRUE )
```

```
##         0.25
## WORDSUM    5
```

So, at least 25% of the United States is estimated to get five correct answers or worse.

Cross-Tabulations

Use **svyby()** with two factor variables. Note that the first colums are point estimates, followed by standard error estimates. The standard error estimates will have the prefix ("se."):

```
# Distribution of SEX across NATEDUC
svyby(~SEX, ~NATEDUC, gss.design, svymean, na.rm = T, ci = F)
```

```
##                 NATEDUC   SEXMale SEXFemale se.SEXMale se.SEXFemale
## Too Little   Too Little 0.3623152 0.6376848 0.01855661   0.01855661
## About Right About Right 0.3560767 0.6439233 0.03196214   0.03196214
## Too Much       Too Much 0.5872315 0.4127685 0.06262315   0.06262315
```

```
# Distribution of NATEDUC across SEX
# Only showing columns with point estimates
svyby(~NATEDUC, ~SEX, gss.design, svymean, na.rm = T, ci = F)[1:4]
```

```
##           SEX NATEDUCToo Little NATEDUCAbout Right NATEDUCToo Much
## Male     Male         0.6949043          0.2109807      0.09411496
## Female Female         0.7320419          0.2283625      0.03959564
```

Mean Scores across Categories

Again, **svyby()**:

```
svyby(~AGE, ~NATEDUC, gss.design, svymean, na.rm = T)
```

```
##                  NATEDUC      AGE        se
## Too Little   Too Little 44.71573 0.5900224
## About Right About Right 47.22340 0.9025996
## Too Much       Too Much 52.01145 1.6784336
```

Percentile Scores across Categories

The 75th percentile age scores across groups with differing opinions on national education spending:

```
svyby(
    ~ AGE, ~ NATEDUC, gss.design, svyquantile ,
    0.75, ci = T, keep.var = T,
    na.rm = TRUE)
```

```
##                  NATEDUC AGE        se
## Too Little   Too Little  58 0.5102135
## About Right About Right  60 1.0204269
## Too Much       Too Much  61 2.5406758
```

Regression Analysis

Linear Regression

To execute a linear regression:

```
model.1 <-
    svyglm(AGE ~ NATEDUC + SEX , gss.design)

summary(model.1)
```

```
##
## Call:
## svyglm(formula = AGE ~ NATEDUC + SEX, gss.design)
##
## Survey design:
## svydesign(ids = ~VPSU, strata = ~VSTRAT, data = DAT, weights = ~WTSSALL,
##     nest = T)
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          42.3998     0.9031  46.952  < 2e-16 ***
## NATEDUCAbout Right    2.4863     1.1095   2.241  0.02863 *
## NATEDUCToo Much       8.1033     1.7073   4.746 1.26e-05 ***
## SEXFemale             3.6336     1.0598   3.429  0.00108 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 296.335)
##
## Number of Fisher Scoring iterations: 2
```

Interpretation is the same as standard linear models. Those who think that the government spends "too much" on education are on average 8.1 years older than their counterparts who believe that the government spends "too little", net of gender. Women are on average 3.6 years older, net of opinions on governmental education spending.

Logit Regression

```
model.2 <-
  svyglm(SEX ~ AGE + NATEDUC,
         family = "binomial", gss.design)
summary(model.2)
```

```
##
## Call:
## svyglm(formula = SEX ~ AGE + NATEDUC, family = "binomial", gss.design)
##
## Survey design:
## svydesign(ids = ~VPSU, strata = ~VSTRAT, data = DAT, weights = ~WTSSALL,
##     nest = T)
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.014473   0.180191   0.080 0.936240
## AGE                  0.012431   0.003581   3.472 0.000948 ***
## NATEDUCAbout Right -0.004887   0.164810  -0.030 0.976440
## NATEDUCToo Much    -1.007007   0.251556  -4.003 0.000170 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1.005889)
##
## Number of Fisher Scoring iterations: 4
```

As with the standard logit, the dependent variable is an odds ratio. You can interpret the effect by exponentiating the odds:

```
exp(model.2$coefficients)-1
```

```
##        (Intercept)                AGE NATEDUCAbout Right
##        0.014578424        0.012508701       -0.004875042
##    NATEDUCToo Much
##       -0.634689383
```

The model suggests that, for each additional year of age, a data point has +1.4% higher odds of pertaining to a female (net of opinions on education funding). Those who believe that the government spends "too much" on education have 63% lower odds of being female, other factors holding constant.

Visualization

In practice, I generate inferences using the operations above, store the results in an object, and then create visualizations using that object. Let's say that you wanted me to graph college attainment differences by gender:

Create the Results Object

This is a comparison of two discrete variables. This calls for a cross-tabultation, which is conventionally graphed using bar or pie charts. First, we run the cross-tabulation, as outlined above:

```
results.1 <- svyby(~DEGREE, ~SEX, gss.design, svymean, na.rm = T, ci = F)
results.1
```

```
##           SEX DEGREEBelow High School DEGREEHigh School
## Male     Male               0.1242347         0.5270168
## Female Female               0.1182799         0.5081559
##        DEGREEJunior College DEGREEBachelors DEGREEGraduate
## Male             0.07030722       0.1797049     0.09873646
## Female           0.07797743       0.1859817     0.10960512
##        se.DEGREEBelow High School se.DEGREEHigh School
## Male                   0.01244929           0.01629705
## Female                 0.01070049           0.01599546
##        se.DEGREEJunior College se.DEGREEBachelors se.DEGREEGraduate
## Male               0.008006702         0.01142566       0.011368663
## Female             0.007747374         0.01381112       0.009576667
```

```
names(results.1)
```

```
##  [1] "SEX"                          "DEGREEBelow High School"
##  [3] "DEGREEHigh School"            "DEGREEJunior College"
##  [5] "DEGREEBachelors"              "DEGREEGraduate"
##  [7] "se.DEGREEBelow High School"   "se.DEGREEHigh School"
##  [9] "se.DEGREEJunior College"      "se.DEGREEBachelors"
## [11] "se.DEGREEGraduate"
```

```
#columns 5 and 6 give us the college graduate point estimates across sex:

results.2 <- results.1[c(5:6)]
#Trims the data set

results.2$COLLEGE <-rowSums(results.2)
#Calculates total college grads

results.2
```

```
##        DEGREEBachelors DEGREEGraduate   COLLEGE
## Male         0.1797049     0.09873646 0.2784414
## Female       0.1859817     0.10960512 0.2955868
```

```
#Graphing Bar Chart
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

```
library(scales)
ggplot(results.2, aes(y = COLLEGE, x = c("Male", "Female"))) +
  geom_bar(stat = 'identity') + ggtitle("College Attainment: Males vs. Females") +
```

```
  scale_y_continuous(breaks = seq(0, 0.3, 0.05), labels = percent_format(accuracy = 1)) +
  xlab("Gender") + ylab("Proportion College-Educated")
```