design weights and probabilities

Peter Lugtig

03 oktober, 2021

Introduction

This is an important exercise. It will introduce inclusion probabilities and design weights as two ways to work with the Horvitz-Thompson estimator (HT), which provides a flexible way to correctly specify your survey design, even when you can't work out the survey design (or when the survey design is very complex). To illustrate why the HT estimator is so useful, we will again use the 'boys' dataset

```
# install.packages("tidyverse")
# install.packages("magrittr")
# install.packages("survey")
# install.packages("sampling")

library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.1
## Warning: package 'ggplot2' was built under R version 4.1.1
## Warning: package 'tibble' was built under R version 4.1.1
## Warning: package 'tidyr' was built under R version 4.1.1
## Warning: package 'readr' was built under R version 4.1.1
## Warning: package 'purrr' was built under R version 4.1.1
## Warning: package 'stringr' was built under R version 4.1.1
## Warning: package 'forcats' was built under R version 4.1.1
```

```
library(magrittr)
```

```
## Warning: package 'magrittr' was built under R version 4.1.1
```

```
library(survey)
```

```
## Warning: package 'survey' was built under R version 4.1.1
```

```
library(sampling)
```

```
## Warning: package 'sampling' was built under R version 4.1.1
```

```
boys <- read_rds("boys.RDS")
```

Working with probabilities & design weights

Whenever we use unequal selection probabilities (e.g. cluster and stratification) we need to specify a 'svydesign()' object that tells R the structure of our sampling design, so that R can compute our outcome statistics per stratum/cluster and then combine them. There is an entirely different way to work with datasets

that have used unequal sampling designs: the HT-estimator uses the individual inclusion probabilities of every case in the dataset.

We will for now use a part of the cluster sampling design we used in week 40, where we draw a two-stage cluster sample. We first draw 2 towns SRS, and then in each town 50 cases using SRS again using the code below. Feel free to run the code with another sampling design (e.g. more clusters or including strata). It is good to practice with this, and a good way is to just alter the basic code I provide

```
# first the sampling bit (repeated from week 40)
boys$id <- 1:nrow(boys)
table(boys$town)
```

```
##
##   1   2   3   4   5
## 191  81 239 161  73
```

```
samplesizestown <- c(table(boys$town))

set.seed(123)
samplec <- sample(5,2)
samplec
```

```
## [1] 3 2
```

```
# removing missing values in town
boys2 <- boys %>%
    filter(!is.na(town))%>%
    arrange(town)
# draw the cluster sample
clustersample <-
    boys2 %>%
    filter(town %in% samplec) %>%
    group_by(town) %>%
    filter(srswor(50, n()) == 1)

# Now we specify the svydesign
# specify the fpc, which now exists of two components:
# the number of clusters, and the size within every sampled cluster
clustersample$ncluster <- 5 # 5 towns in population
clustersample$clustersize <- NA # empty first, then add sample sizes in towns 2 and 3
clustersample$clustersize[clustersample$town == 2] <- nrow(boys2[boys2$town == 2, ])
clustersample$clustersize[clustersample$town == 3] <- nrow(boys2[boys2$town == 3, ])

# or in a tidy way
# clustersample <-
#     boys2 %>%
#     group_by(town) %>%
#     summarize(clustersize = n()) %>%
#     right_join(clustersample1)

# and specify the cluster design
clusterdesign <- svydesign(id = ~town + id,
                           fpc = ~ncluster + clustersize,
                           data = clustersample)
```

Q1

What is the mean for the variable 'wgt' and its standard error?

The Hurvitz Thompson estimator

Because we have drawn the clusters using SRS, the inclusion probabilities of the individual boys in the dataset vary. By specifying the svydesign() object, we tell R to compute the mean separately in every cluster, and then combine the means from the clusters weighted by the cluster size. Can you compute for the individuals in clusters 2 and 3 (within the clusters drawn) what was the inclusion probability?

The probabilities can also be derived from the clusterdesign svydesign object by asking for:

```
clusterdesign[["prob"]]
# exactly the same as manually computed ones
# just asking for:
clusterdesign
# actually also tells us what design we just specified!
```

The central idea of the HT estimator is that instead of specifying the specific clustering and stratification that we have in our sample, we can do a much easier job by simply including the inclusion probability for every case in our sample. Statistics like the mean can be easily computed when the inverse of the inclusion probability is used as a weighting factor for every individual in the dataset. Rather than computing the mean per stratum/cluster, the HT estimator: 1. sums all the weighted individual values, 2. and then divides by the sum of inclusion probabilities.

Using the HT estimator in R is easy. We simply use the inclusion probabilities in the probs=~ command. See the code below.

##Q2: What is the survey mean for the variable 'wgt' and its standard error under the HT estimator?

```
# HT estimator
clustersample$fpc <- 748
HT1<- svydesign(id=~town, probs =~prob, data=clustersample)
svymean(~wgt,design=HT1,deff=T)
```

You should find here that the HT estimator results in a larger standard error. Why is this? Obviously, something happens when we compute the variance using the HT estimator. We will return to this in a bit.

Design weights

Often, public use datasets (like your adopted survey) do not include the specific cluster or stratification indicators, nor the inclusion probabilities, but rather "Design weights". A Design weight is an indicator that tells you "how many population elements" a row in your dataset represents. Calculating design weights is easy: they are the inverse of the inclusion probabilities. We can also use design weights rather than the probabilities in our svydesign object. As soon as we provide either probabilities or weights in the 'weights=~ command, the survey package will switch to use the HT estimator.

##Q3: Check the code below. Do you get the same results using inclusion probabilities and weights?

```
clustersample$designweight <- 1/clustersample$prob
HT2 <- svydesign(id=~town,weights=~designweight,data=clustersample)
svymean(~wgt,design=HT2,deff=T)
# equivalent!
```

The tricky estimation of variances under HT-estimators

Stuart (1984) showed you that the variance of a cluster or stratified design can be computed as the weighted average of variance in every stratum/cluster. Horvitz Thompson works with individual observations, so how

do we define the variance now?!

This course will not discuss methods to estimate the variance (of e.g. the mean) under all conditions in detail, but below I will show some ways to compute a variance for HT estimators.

There are over 50 different algorithms to compute the variance for HT type estimators, and there is no agreement over which method works (generally) better.

The most common way to compute the variance is by Taylor series expansion (review your slides from Fundamentals on how this works). The idea here is that we can approximate the variance total by estimating the variance of an unknown distribution (the inclusion probabilities) by fitting a series of polynomials (x, X^2, X^3, X^4, etc.), and then derive the total variance by averaging the variance over the fitted function.

Another (perhaps more modern) way is to use Bootstrapping. You will learn more about this later this semester, but the general idea here is that you can resample (with replacement!) from the actual sample to get a large new number of samples (e.g 1000 new replicate samples). After this, you get 1000 point estimates for every individual in your dataset, which you can then use to estimate the total variance. This also means that you get 100 inclusion probabilities and design weights per case. The survey package can however work with such " bootstrap replicate weights", and pool the estimates for you.

If you want to know more about different versions of the Taylor expansions available, the survey package includes a few standard algorithmns (see below). If you would like to read more (this will not be tested), you may read:

Mattei & Tille (2005) Evaluation of Variance Approximations and Estimators in Maximum Entropy Sampling with Unequal Probability and Fixed Sample Size, Journal of Official Statistics, Vol. 21, No. 4, 2005, pp. 543–570

The differences between these algorithms is usually quite small.

```
HT3 <- svydesign(id=~town,  probs=~prob, data=clustersample, pps="brewer")
HT4 <- svydesign(id=~town,  probs=~prob, data=clustersample, pps="overton")
HT5 <- svydesign(id=~town,  probs=~prob, data=clustersample, variance="HT")

svymean(~wgt,design=HT3,deff=T) #deff=5.97 # equivalent
```

```
##        mean      SE   DEff
## wgt 38.8666  6.0323 5.9762
```

```
svymean(~wgt,design=HT4,deff=T) #deff=4.98 # some underestimation
```

```
##        mean      SE   DEff
## wgt 38.8666  5.5113 4.9883
```

```
svymean(~wgt,design=HT5,deff=T) #deff=5.97 # equivalent
```

```
##        mean      SE   DEff
## wgt 38.8666  6.0323 5.9762
```

```
# all designs overestimate the variance compared to the normal cluster design
# this is quite common with the HT estimator: it is ' conservative' in estimating standard errors
```

why use a HT estimator if variance estimation is so tricky?

In a simple sample design, you don't need to use the HT estimator. But in real life sampling designs are complex. For example, you could in real life situations encounter situations where some cluster and/or stratum sizes are small, and others large. In such situations, you will have very different group sizes, and quite likely end up with subgroups of just a few or even 1 case. Computing variances for such small groups is risky (as the variance can become very small or even 0), and if you somehow end up with just 1 case in a cluster/stratum, the variance is undefined!

The example below illustrates this. We first add a bit of random noise to the inclusion probbilities, so that we get group sizes of 1 (and variances cannot be calculated)

```
# add some noise to the inclusion probabilities
require(MASS)
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
clustersample$prob2 <- rnorm(n=nrow(clustersample),mean=clustersample$prob,sd=.02)
table(clustersample$prob2) # all diifferent probabilities now.
# HT estimator
HT6 <- svydesign(id=~town, probs=~prob2, data=clustersample)
svymean(~wgt,design=HT6,deff=T) #deff=5.4 # small difference, due to small diff in standard error.
```

The main take-away of all of this? The HT estimator is flexible, and easy to use under any sampling design. You however give

#end of file

— END OF DOCUMENT —