INTRODUCTION TO SURVEY DATA ANALYSIS THROUGH STATISTICAL PACKAGES

Hukum Chandra
Indian Agricultural Statistics Research Institute, New Delhi-110012

1. INTRODUCTION

A sample survey is a process for collecting data on a sample of observations which are selected from the population of interest using a probability-based sample design. In sample surveys, certain methods are often used to improve the precision and control the costs of survey data collection. These methods introduce a complexity to the analysis, which must be accounted for in order to produce unbiased estimates and their associated levels of precision. This write up provides a brief introduction to the impact these design complexities have on the sampling variance and then summarizes the analysis on sample survey data using softwares.

2. COMPLEX SAMPLE DESIGNS

Statistical methods for estimating population parameters and their associated variances are based on assumptions about the characteristics and underlying distribution of the observations. Statistical methods in most general-purpose statistical software tacitly assume that the data meet certain assumptions. Among these assumptions are that the observations were selected independently and that each observation had the same probability of being selected. Data collected through surveys often have sampling schemes that deviate from these assumptions. For logistical reasons, samples are often clustered geographically to reduce costs of administering the survey, and it is not unusual to sample households, then subsample families and/or persons within selected households. In these situations, sample members are not selected independently, nor are their responses likely to be independently distributed.

In addition, a common survey sampling practice is to oversample certain population subgroups to ensure sufficient representation in the final sample to support separate analyses. This is particularly common for certain policy-relevant subgroups, such as ethnic and racial minorities, the poor, the elderly, and the disabled. In this situation, sample members do not have equal probabilities of selection. Adjustments to sampling weights (the inverse of the probability of selection) to account for nonresponse, as well as other weighting adjustments (such as poststratification to known population totals), further exacerbate the disparity in the weights among sample members.

In brief, the complications in a complex survey sample result from following:

- **Stratification**- Dividing the population into relatively homogenous groups (strata) and sampling a predetermined number from each stratum will increase precision for a given sample size.

- **Clustering**- Dividing the population into groups and sampling from a random subset of these groups (eg geographical locations) will decrease precision for a given sample size but often increase precision for a given cost.
- **Unequal sampling** - Sampling small subpopulations more heavily will tend to increase precision relative to a simple random sample of the same size.

- **Finite population** - Sampling all of a population or stratum results in an estimate with no variability, and sampling a substantial fraction of a stratum results in decreased variability in comparison to a sample from an infinite population. I have described these in terms of their effect on the design of the survey.

- **Weighting** - When units are sampled with unequal probability it is necessary to give them correspondingly unequal weights in the analysis. The inverse-probability weighting has generally the same effect on point estimates as the more familiar inverse-variance weighting, but very different effects on standard errors.

Most standard statistical procedures in software packages commonly used for data analysis do not allow the analyst to take most of these properties of survey data into account unless specialized survey procedures are used. That is standard methods of statistical analysis assume that survey data arise from a *simple random sample* of the target population. Little attention is given to characteristics often associated with survey data, including missing data, unequal probabilities of observation, stratified multistage sample designs, and measurement errors. Failure to do so can have an important impact on the results of all types of analysis, ranging from simple descriptive statistics to estimates of parameters of multivariate models.

### 3. IMPACT OF COMPLEX SAMPLE DESIGN ON SAMPLING VARIANCE

Because of these deviations from standard assumptions about sampling, such survey sample designs are often referred to as complex. While stratification in the sampling process can decrease the sampling variance, clustering and unequal selection probabilities generally increase the sampling variance associated with resulting estimates. Not accounting for the impact of the complex sample design can lead to an underestimate of the sampling variance associated with an estimate. So while standard software packages can generally produce an unbiased weighted survey estimate, it is quite possible to have an underestimate of the precision of such an estimate when using one of these packages to analyze survey data.

That is, analyzing a stratified sample as if it were a simple random sample will *overestimate* the standard errors, analyzing a cluster sample as if it were a simple random sample will usually *underestimate* the standard errors, as will analyzing an unequal probability sample as if it were a simple random sample.

The magnitude of this effect on the variance is commonly measured by what is known as the design effect. The design effect is the sampling variance of an estimate, accounting for the complex sample design, divided by the sampling variance of the same estimate, assuming a sample of equal size had been selected as a simple random sample. A design effect of unity indicates that the design had no impact on the variance of the estimate. A design effect greater than one indicates that the design has increased the variance, and a design effect less than one indicates that the design actually decreased the variance of the estimate. The design effect can be used to determine the effective sample size, simply by dividing the nominal sample size by the design effect. The effective sample size gives the number of observations that would yield an equivalent level of precision from an independent and identically distributed (iid) sample.

### 4. SOFTWARE PACKAGES FOR SURVEY DATA ANALYSIS
Introduction to Survey Data Analysis …

Several packages are available to the public designed specifically for use with sample survey data. However, in this lecture I will discuss detail **Software R** for analyzing complex surveys. The survey functions for R were contributed by Thomas Lumley, Department of Biostatistics, University of Washington, USA.

**Types of designs that can be accommodated**

- Designs incorporating stratification, clustering, and possibly multistage sampling, allowing unequal sampling probabilities or weights.
- Simple two-phase designs
- Multiply-imputed data

**Types of estimates and statistical analyses that can be done in R**

- Mean, Totals, Quantiles, Variance, Tables, Ratios,
- Generalized linear models (e.g. linear regression, logistic regression etc.)
- Proportional hazards models
- Proportional odds and other cumulative link models
- Survival curves
- Post-stratification, raking, and calibration
- Tests of association in two-way tables

**Restrictions on number of variables or observations:** Only those due to limitations of available memory or disk capacity.

**Variance estimation methods:** Taylor series linearization and replication weighting.

**Platforms on which the software can be run**

- Intel computers with Windows 2000 or better
- Mac OS X 10.3 or later
- Linux
- Most Unix systems.

**Pricing and terms:** Free download. R is updated about twice per year and the survey package is updated as needed. For information on R see [http://www.r-project.org/](http://www.r-project.org/).

**5. IMPLEMENTATION OF SURVEY PACKAGE IN R**

First install survey package. The command **svydesign** in **library (survey)** is used for survey data analysis in R, described as below.
svydesign \((id=-1, strata=-\text{stype}, weights=-\text{pw}, data=\text{apistrat}, fpc=-\text{fpc})\)

where different arguments of function `svydesign()` are

<table>
<thead>
<tr>
<th>Argument</th>
</tr>
</thead>
<tbody>
<tr>
<td>ids</td>
</tr>
<tr>
<td>probs</td>
</tr>
<tr>
<td>strata</td>
</tr>
<tr>
<td>variables</td>
</tr>
<tr>
<td>fpc</td>
</tr>
<tr>
<td>weights</td>
</tr>
<tr>
<td>data</td>
</tr>
<tr>
<td>nest</td>
</tr>
<tr>
<td>check.strata</td>
</tr>
</tbody>
</table>

The `svydesign` object combines a data frame and all the survey design information needed to analyse it. These objects are used by the survey modelling and summary functions. The \texttt{id} argument is always required, the \texttt{strata}, \texttt{fpc}, \texttt{weights} and \texttt{probs} arguments are \texttt{optional}. If these variables are specified they must \texttt{not} have any missing values.

By default, \texttt{svydesign} assumes that all PSUs, even those in different strata, have a unique value of the \texttt{id} variable. This allows some data errors to be detected. If your PSUs reuse the same identifiers across strata then set \texttt{nest}=\texttt{TRUE}.

The \textit{finite population correction} (\texttt{fpc}) is used to reduce the variance when a substantial fraction of the total population of interest has been sampled. It may not be appropriate if the target of inference is the process generating the data rather than the statistics of a particular finite population.

The finite population correction can be specified either as the total population size in each stratum or as the fraction of the total population that has been sampled. In either case the relevant population size is the sampling units. That is, sampling 100 units from a population stratum of size 500 can be specified as 500 or as \(100/500=0.2\).

If population sizes are specified but not sampling probabilities or weights, the sampling probabilities will be computed from the population sizes assuming simple random sampling within strata.
For multistage sampling the id argument should specify a formula with the cluster identifiers at each stage. If subsequent stages are stratified strata should also be specified as a formula with stratum identifiers at each stage. The population size for each level of sampling should also be specified in fpc. If fpc is not specified then sampling is assumed to be with replacement at the top level and only the first stage of cluster is used in computing variances. If fpc is specified but for fewer stages than id, sampling is assumed to be complete for subsequent stages. The variance calculations for multistage sampling assume simple or stratified random sampling within clusters at each stage except possibly the last.

If the strata with one only PSU are not self-representing (or they are, but svydesign cannot tell based on fpc) then the handling of these strata for variance computation is determined by options("survey.lonely.psu").

Example - Read the api data - Academic Performance Index (api) is computed for all California schools. The full population data in api are a data frame with 6194 observations on the 37 variables. Read api data available in survey package

```r
data(api)  #This load the api population data api

dim(api)  # Shows the dimension of the data set
```

The details of 37 variables are

1. cds Unique identifier
2. stype Elementary/Middle/Hi gh School
3. name School name (15 characters)
4. sname School name (40 characters)
5. snum School number
6. dname District name
7. dnum District number
8. cname County name
9. cnum County number
10. flag reason for missing data
11. pcttest percentage of students tested
12. api00 API in 2000
13. api99 API in 1999
14. target target for change in API
15. growth Change in API
16. sch.wide Met school-wide growth target?
17. comp.imp Met Comparable Improvement target
18. both Met both targets
19. awards Eligible for awards program
20. meals Percentage of students eligible for subsidized meals
21. ell `English Language Learners' (percent)
22. yr.rnd Year-round school
23. mobility percent of students for whom this is the first year at the school
24. acs.k3 average class size years K-3
25. acs.46 average class size years 4-6
26. acs.core Number of core academic courses
27. pct.resp percent where parental education level is known
28. not.hsg percent parents not high-school graduates
29. hsg percent parents who are high-school graduates
30. some.col percent parents with some college
31. col.grad percent parents with college degree
32. grad.sch percent parents with postgraduate education
33. avg.ed average parental education level
34. full percent fully qualified teachers
35. emer percent teachers with emergency qualifications
36. enroll number of students enrolled
37. api.stu number of students tested.

Type `summary(apipop)` and see what you get?

The other data sets contain additional variables pw for sampling weights and fpc to compute finite population corrections to variance. `apipop` is the entire population, `apiclus1` is a cluster sample of school districts, `apistrat` is a sample stratified by stype, and `apiclus2` is a two-stage cluster sample of schools within districts. The sampling weights in `apiclus1` are incorrect (the weight should be 757/15) but are as obtained from UCLA. Data were obtained from the survey sampling help pages of UCLA Academic Technology Services, at


The API program and original data files are at http://api.cde.ca.gov/

# api00 is API in 2000

    mean (api00)
    [1] 664.7126

# enroll is number of students enrolled

    sum (apipop$enroll, na.rm=TRUE)
    [1] 3811472

Here `na.rm=TRUE` means –logical, Should missing values be removed?

**Specifying a complex survey design – use function svydesign ()**

[i]  **Stratified sample**

Here we use data set apistrat, see `dim(apistrat), c(apistrat[1,]), attach(apistrat) commands etc.

`dstrat<- svydesign(id=~1,strata=~stype, weights=~pw, data=apistrat, fpc=~fpc)`

**summary(dstrat)**

<table>
<thead>
<tr>
<th>Stratified Independent Sampling design</th>
</tr>
</thead>
<tbody>
<tr>
<td>svydesign(id = ~1, strata = ~stype, weights = ~pw, data = apistrat, fpc = ~fpc)</td>
</tr>
</tbody>
</table>

Probabilities:

<table>
<thead>
<tr>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
</table>
Stratum Sizes:

<table>
<thead>
<tr>
<th></th>
<th>E</th>
<th>H</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>obs</td>
<td>100</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

design.PSU

<table>
<thead>
<tr>
<th></th>
<th>E</th>
<th>H</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>50</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>

actual.PSU

<table>
<thead>
<tr>
<th></th>
<th>E</th>
<th>H</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>50</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>

Population stratum sizes (PSUs):

<table>
<thead>
<tr>
<th></th>
<th>E</th>
<th>M</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>4421</td>
<td>1018</td>
<td>755</td>
<td></td>
</tr>
</tbody>
</table>

Data variables:

1] "cds" "stype" "name" "sname" "snum" "dname"
2] "dnum" "cnum" "nnum" "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" "pw" "fpc"

Some functions used to compute means, variances, ratios and totals for data from complex surveys are as follows.

svymean() and svytotal() functions are use to extract mean and total estimate along with their standard error, specified as below.

svymean(x, design, na.rm=FALSE, deff=FALSE,...)

svytotal(x, design, na.rm=FALSE, deff=FALSE,...)

**Arguments**

x A formula, vector or matrix
design survey.design or svyrep.design object
na.rm Should cases with missing values be dropped?
rho parameter for Fay's variance estimator in a BRR design
return.replicates Return the replicate means?
deff Return the design effect
object The result of one of the other survey summary functions
Introduction to Survey Data Analysis

quietly
Don't warn when there is no design effect computed

estimate.only
Don't compute standard errors (useful when svyvar is used to estimate the design effect)

names
vector of character strings

Also see
Svyvar (x, design, na.rm=FALSE,...)
svyratio (x, design, na.rm=FALSE,...)
svyquantile (x, design, na.rm=FALSE,...)

svymean(~api00, dstrat)

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>api00</td>
<td>662.29</td>
<td>9.4089</td>
</tr>
</tbody>
</table>

svymean(~api00, dstrat, deff=TRUE)

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>SE</th>
<th>DEff</th>
</tr>
</thead>
<tbody>
<tr>
<td>api00</td>
<td>662.29</td>
<td>9.4089</td>
<td>1.2045</td>
</tr>
</tbody>
</table>

svytotal(~enroll, dstrat, na.rm=TRUE)

<table>
<thead>
<tr>
<th></th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>enroll</td>
<td>3687178</td>
</tr>
</tbody>
</table>

#stratified sample, Now try these code for your self
dstrat<-svydesign(id=~1, strata=~stype, weights=~pw, data=apistrat, fpc=~fpc)
summary(dstrat)
svymean(~api00, dstrat)
svyquantile(~api00, dstrat, c(.25,.5,.75))
svyvar(~api00, dstrat)
svytotal(~enroll, dstrat)
svyratio(~api.stu, ~enroll, dstrat)
# coefficients of variation
cv(svytotal(~enroll,dstrat))

[ii] One-stage cluster sample
dclus1<-svydesign(id=~dnum, weights=~pw, data=apiclus1, fpc=~fpc)
summary(dclus1)
svymean(~api00, dclus1, deff=TRUE)
svymean(~factor(stype),dclus1)
svymean(~interaction(stype, comp.imp), dclus1)
svyquantile(~api00, dclus1, c(.25,.5,.75))
svyvar(~api00, dclus1)
svytotal(~enroll, dclus1, deff=TRUE)
svyratio(~api.stu, ~enroll, dclus1)

**summary(dclus1)**

1 - level Cluster Sampling design

With (15) clusters.

svydesign(id = ~dnum, weights = ~pw, data = apiclus1, fpc = ~fpc)

Probabilities:

<table>
<thead>
<tr>
<th></th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.02954</td>
<td>0.02954</td>
<td>0.02954</td>
<td>0.02954</td>
<td>0.02954</td>
<td>0.02954</td>
</tr>
</tbody>
</table>

Population size (PSUs): 757

Data variables:

[1]  "cids" "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"

svymean(~api00, dclus1)

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>api00</td>
<td>644.17</td>
<td>23.542</td>
</tr>
</tbody>
</table>

svytotal(~enroll, dclus1, na.rm=TRUE)

<table>
<thead>
<tr>
<th></th>
<th>total</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>enroll</td>
<td>3404940</td>
<td>932235</td>
</tr>
</tbody>
</table>

[iii] Two-stage cluster sample

dclus2<-svydesign(id=~dnum+snum, fpc=~fpc1+fpc2, data=apiclus2)

**summary(dclus2)**

2 - level Cluster Sampling design

With (40, 126) clusters.

svydesign(id = ~dnum + snum, fpc = ~fpc1 + fpc2, data = apiclus2)

Probabilities:

<table>
<thead>
<tr>
<th></th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7: Introduction to Survey Data Analysis …

Population size (PSUs): 757

Data variables:

[1] "cds"  "stype"  "name"  "sname"  "snum"  "dname"
[7] "dnum"  "cname"  "cnum"  "flag"  "pcttest"  "api00"
[13] "api99"  "target"  "growth"  "sch.width"  "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"  "pw"  "fpc1"  "fpc2"

\texttt{svymean(~api00, dclus2)}

<table>
<thead>
<tr>
<th>mean</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>670.81</td>
<td>30.099</td>
</tr>
</tbody>
</table>

\texttt{svytotal(~enroll, dclus2, na.rm=TRUE)}

<table>
<thead>
<tr>
<th>enroll</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2639273</td>
<td>799638</td>
</tr>
</tbody>
</table>

[iv] Two-stage `with replacement`

\texttt{dclus2wr<-svydesign(id=~dnum+snum, weights=~pw, data=apiclus2)}

\texttt{summary(dclus2wr)}

2 - level Cluster Sampling design (with replacement)

With (40, 126) clusters.

svydesign(id = ~dnum + snum, weights = ~pw, data = apiclus2)

Probabilities:

<table>
<thead>
<tr>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.003669</td>
<td>0.037740</td>
<td>0.052840</td>
<td>0.042390</td>
<td>0.052840</td>
<td>0.052840</td>
</tr>
</tbody>
</table>

Data variables:

[1] "cds"  "stype"  "name"  "sname"  "snum"  "dname"
[7] "dnum"  "cname"  "cnum"  "flag"  "pcttest"  "api00"
[13] "api99"  "target"  "growth"  "sch.width"  "comp.imp"  "both"
[19] "awards"  "meals"  "ell"  "yr.rnd"  "mobility"  "acs.k3"
svymean(~api00, dclus2wr)

<table>
<thead>
<tr>
<th>mean</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>670.81</td>
<td>30.712</td>
</tr>
</tbody>
</table>

svytotal(~enroll, dclus2wr, na.rm=TRUE)

<table>
<thead>
<tr>
<th>total</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2639273</td>
<td>820261</td>
</tr>
</tbody>
</table>

REFERENCE