```
make pkg-geepack_0.2-4
```

If there is no error, the Windows binary `geepack.zip` will be created in the `WinRlibs` subdirectory, which is then ready to be shipped to a Windows machine for installation.

We can easily build bundled packages as well. For example, to build the packages in bundle `VR`, we place the source `VR_7.0-11.tar.gz` into the `pkgsrc` subdirectory, and then do

```
make bundle-VR_7.0-11
```

The Windows binaries of packages `MASS`, `class`, `nnet`, and `spatial` in the `VR` bundle will appear in the `WinRlibs` subdirectory.

This Makefile assumes a tarred and gzipped source for an R package, which ends with "`.tar.gz`". This is usually created through the `R CMD build` command. It takes the version number together with the package name.

The Makefile

The associated Makefile is used to automate many of the steps. Since many Linux distributions come with the **make** utility as part of their installation, it hopefully will help rather than confuse people cross-compiling R. The Makefile is written in a format similar to shell commands in order to show what exactly each step does.

The commands can also be cut-and-pasted out of the Makefile with minor modification (such as, change $$ to $ for environmental variable names), and run manually.

Possible pitfalls

We have very little experience with cross-building packages (for instance, Matrix) that depend on external libraries such as atlas, blas, lapack, or Java libraries. Native Windows building, or at least a substantial amount of testing, may be required in these cases. It is worth experimenting, though!

Acknowledgments

We are grateful to Ben Bolker, Stephen Eglen, and Brian Ripley for helpful discussions.

Jun Yan
University of Wisconsin–Madison, U.S.A.
jyan@stat.wisc.edu

A.J. Rossini
University of Washington, U.S.A.
rossini@u.washington.edu

Analysing Survey Data in R

by Thomas Lumley

Introduction

Survey statistics has always been a somewhat specialised area due in part to its view of the world. In the rest of the statistical world data are random and we model their distributions. In traditional survey statistics the population data are fixed and only the sampling is random. The advantage of this 'design-based' approach is that the sampling, unlike the population, is under the researcher's control.

The basic question of survey statistics is

> If we did exactly this analysis on the whole population, what result would we get?

If individuals were sampled independently with equal probability the answer would be very straightforward. They aren't, and it isn't.

To simplify the operations of a large survey it is routine to sample in clusters. For example, a random sample of towns or counties can be chosen and then individuals or families sampled from within those areas. There can be multiple levels of cluster sampling, eg, individuals within families within towns. Cluster sampling often results in people having an unequal chance of being included, for example, the same number of individuals might be surveyed regardless of the size of the town.

For statistical efficiency and to make the results more convincing it is also usual to stratify the population and sample a prespecified number of individuals or clusters from each stratum, ensuring that all strata are fairly represented. Strata for a national survey might divide the country into geographical regions and then subdivide into rural and urban. Ensuring that smaller strata are well represented may also involve sampling with unequal probabilities.

Finally, unequal probabilities might be deliberately used to increase the representation of groups that are small or particularly important. In US health surveys there is often interest in the health of the poor and of racial and ethnic minority groups, who might thus be oversampled.

The resulting sample may be very different in structure from the population, but it is different in ways that are precisely known and can be accounted for in the analysis.

Weighted estimation

The major technique is inverse-probability weighting to estimate population totals. If the probability of sampling individual i is π_i and the value of some variable or statistic for that person is Y_i then an unbiased estimate of the population total is

$$\sum_i \frac{1}{\pi_i} Y_i$$

regardless of clustering or stratification. This is called the Horvitz–Thompson estimator (Horvitz & Thompson, 1951). Inverse-probability weighting can be used in an obvious way to compute population means and higher moments. Less obviously, it can be used to estimate the population version of loglikelihoods and estimating functions, thus allowing almost any standard models to be fitted. The resulting estimates answer the basic survey question: they are consistent estimators of the result that would be obtained if the same analysis was done to the whole population.

It is important to note that estimates obtained by maximising an inverse-probability weighted loglikelihood are not in general maximum likelihood estimates under the model whose loglikelihood is being used. In some cases semiparametric maximum likelihood estimators are known that are substantially more efficient than the survey-weighted estimators if the model is correctly specified. An important example is a two-phase study where a large sample is taken at the first stage and then further variables are measured on a subsample.

Standard errors

The standard errors that result from the more common variance-weighted estimation are incorrect for probability weighting and in any case are model-based and so do not answer the basic question. To make matters worse, stratified and clustered sampling also affect the variance, the former decreasing it and the latter (typically) increasing it relative to independent sampling.

The formulas for standard error estimation are developed by first considering the variance of a sum or mean under independent unequal probability sampling. This can then be extended to cluster sampling, where the clusters are independent, and to stratified sampling by adding variance estimates from each stratum. The formulas are similar to the 'sandwich variances' used in longitudinal and clustered data (but not quite the same).

If we index strata by s, clusters by c and observations by i then

$$\hat{\mathrm{var}}[S] = \sum_s \frac{n_s}{n_s - 1} \sum_{c,i} \left(\frac{1}{\pi_{sci}} \left(Y_{sci} - \bar{Y}_s \right) \right)^2$$

where S is the weighted sum, $\bar{Y}_s$ are weighted stratum means, and n_s is the number of clusters in stratum s.

Standard errors for statistics other than means are developed from a first-order Taylor series expansion, that is, they use the same formula applied to the mean of the estimating functions. The computations for variances of sums are performed in the `svyCprod` function, ensuring that they are consistent across the survey functions.

Subsetting surveys

Estimates on a subset of a survey require some care. Firstly, it is important to ensure that the correct weight, cluster and stratum information is kept matched to each observation. Secondly, it is not correct simply to analyse a subset as if it were a designed survey of its own.

The correct analysis corresponds approximately to setting the weight to zero for observations not in the subset. This is not how it is implemented, since observations with zero weight still contribute to constraints in R functions such as `glm`, and they still take up memory. Instead, the `survey.design` object keeps track of how many clusters (PSUs) were originally present in each stratum and `svyCprod` uses this to adjust the variance.

Example

The examples in this article use data from the National Health Interview Study (NHIS 2001) conducted by the US National Center for Health Statistics. The data and documentation are available from `http://www.cdc.gov/nchs/nhis.htm`. I used NCHS-supplied SPSS files to read the data and then `read.spss` in the `foreign` package to load them into R. Unfortunately the dataset is too big to include these examples in the `survey` package — they would increase the size of the package by a couple of orders of magnitude.

The survey package

In order to keep the stratum, cluster, and probability information associated with the correct observations the `survey` package uses a `survey.design` object created by the `svydesign` function. A simple example call is

```
imdsgn<-svydesign(id=~PSU,strata=~STRATUM,
            weights=~WTFA.IM,
            data=immunize,
            nest=TRUE)
```

The data for this, the immunization section of NHIS, are in the data frame immunize. The strata are specified by the STRATUM variable, and the inverse probability weights are in the WTFA.IM variable. The statement specifies only one level of clustering, by PSU. The nest=TRUE option asserts that clusters are nested in strata so that two clusters with the same psuid in different strata are actually different clusters.

In fact there are further levels of clustering and it would be possible to write

```
imdsgn<-svydesign(id=~PSU+HHX+FMX+PX,
            strata=~STRATUM,
            weights=~WTFA.IM,
            data=immunize,
            nest=TRUE)
```

to indicate the sampling of households, families, and individuals. This would not affect the analysis, which depends only on the largest level of clustering. The main reason to provide this option is to allow sampling probabilities for each level to be supplied instead of weights.

The strata argument can have multiple terms: eg strata= region+rural specifying strata defined by the interaction of region and rural, but NCHS studies provide a single stratum variable so this is not needed. Finally, a finite population correction to variances can be specified with the fpc argument, which gives either the total population size or the overall sampling fraction of top-level clusters (PSUs) in each stratum.

Note that variables to be looked up in the supplied data frame are all specified by formulas, removing the scoping ambiguities in some modelling functions.

The resulting survey.design object has methods for subscripting and na.action as well as the usual print and summary. The print method gives some descriptions of the design, such as the number of largest level clusters (PSUs) in each stratum, the distribution of sampling probabilities and the names of all the data variables. In this immunisation study the sampling probabilities vary by a factor of 100, so ignoring the weights may be very misleading.

Analyses

Suppose we want to see how many children have had their recommended polio immmunisations recorded. The command

```
 svytable(~AGE.P+POLCT.C, design=imdsgn)
```

produces a table of number of polio immunisations by age in years for children with good immunisation data, scaled by the sampling weights so that it corresponds to the US population. To fit the table more easily on the page, and since three doses is regarded as sufficient we can do

```
> svytable(~AGE.P+pmin(3,POLCT.C),design=imdsgn)
     pmin(3, POLCT.C)
AGE.P      0      1      2       3
    0 473079 411177 655211  276013
    1 162985  72498 365445  863515
    2 144000  84519 126126 1057654
    3  89108  68925 110523 1123405
    4 133902 111098 128061 1069026
    5 137122  31668  19027 1121521
    6 127487  38406  51318  838825
```

where the last column refers to 3 or more doses. For ages 3 and older, just over 80% of children have received 3 or more doses and roughly 10% have received none.

To examine whether this varies by city size we could tabulate

```
 svytable(~AGE.P+pmin(3,POLCT.C)+MSASIZEP,
           design=imdsgn)
```

where MSASIZEP is the size of the 'metropolitan statistical area' in which the child lives, with categories ranging from 5,000,000+ to under 250,000, and a final category for children who don't live in a metropolitan statistical area. As MSASIZEP has 7 categories the data end up fairly sparse. A regression model might be more useful.

Regression models

The svyglm and svycoxph functions have similar syntax to their non-survey counterparts, but use a design rather than a data argument. For simplicity of implementation they do require that all variables in the model be found in the design object, rather than floating free in the calling environment.

To look at associations with city size we fit the logistic regression models in Figure 1. The resulting svyglm objects have a summary method similar to that for glm. There isn't a consistent association, but category 2: cities from 2.5 million to 5 million people, does show some evidence of lower vaccination rates. Similar analyses using family income don't show any real evidence of an assocation.

General weighted likelihood estimation

The svymle function maximises a specified weighted likelihood. It takes as arguments a loglikelihood function and formulas or fixed values for each parameter in the likelihood. For example, a linear regression could be implemented by the code in Figure 2. In this example the log=TRUE argument is passed to dnorm and is the only fixed argument. The other two arguments, mean and sd, are specified by

```
svyglm(I(POLCT.C>=3)~factor(AGE.P)+factor(MSASIZEP), design=imdsgn, family=binomial)
svyglm(I(POLCT.C>=3)~factor(AGE.P)+MSASIZEP,  design=imdsgn, family=binomial)
```

Figure 1: Logistic regression models for vaccination status

```
gr <- function(x,mean,sd,log)
     dm <- 2*(x - mean)/(2*sd^2)
     ds <- (x-mean)^2*(2*(2 * sd))/(2*sd^2)^2 - sqrt(2*pi)/(sd*sqrt(2*pi))
     cbind(dm,ds)

m2 <- svymle(loglike=dnorm,gradient=gr, design=dxi,
             formulas=list(mean=y~x+z, sd=~1),
             start=list(c(2,5,0),  c(4)),
             log=TRUE)
```

Figure 2: Weighted maximum likelihood estimation

formulas. Exactly one of these formulas should have a left-hand side specifying the response variable.

It is necessary to specify the gradient in order to get survey-weighted standard errors. If the gradient is not specified you can get standard errors based on the information matrix. With independent weighted sampling the information matrix standard errors will be accurate if the model is correctly specified. As is always the case with general-purpose optimisation it is important to have reasonable starting values; you could often get them from an analysis that ignored the survey design.

Extending the survey package

It is easy to extend the survey package to include any new class of model where weighted estimation is possible. The svycoxph function shows how this is done.

1. Create a call to the underlying model (coxph), adding a weights argument with weights $1/\pi_i$ (or a rescaled version).
2. Evaluate the call
3. Extract the one-step delta-betas Δ_i or compute them from the information matrix I and score contributions U_i as $\Delta = I^{-1}U_i$
4. Pass Δ_i and the cluster, strata and finite population correction information to svyCprod to compute the variance.
5. Add a "svywhatever" class to the object and a copy of the design object
6. Override the print and print.summary methods to include print(x$design, varnames=FALSE, design.summaries=FALSE)
7. Override any likelihood extraction methods to fail

It may also be necessary to override other methods such as residuals, as is shown for Pearson residuals in residuals.svyglm.

A second important type of extension is to add prediction methods for small-area extrapolation of surveys. That is, given some variables measured in a small geographic area (a new cluster), predict other variables using the regression relationship from the whole survey. The predict methods for survey regression methods current give predictions only for individuals.

Extensions in third direction would handle different sorts of survey design. The current software cannot handle surveys where strata are defined within clusters, partly because I don't know what the right variance formula is.

Summary

Analysis of complex sample surveys used to require expensive software on expensive mainframe hardware. Computers have advanced sufficiently for a reasonably powerful PC to analyze data from large national surveys such as NHIS, and R makes it possible to write a useful survey analysis package in a fairly short time. Feedback from practising survey statisticians on this package would be particularly welcome.

References

Horvitz DG, Thompson DJ (1951). A Generalization of Sampling without Replacement from a Finite Universe. *Journal of the American Statistical Association*. 47, 663-685.

Thomas Lumley
Biostatistics, University of Washington
tlumley@u.washington.edu