THE WORLD BANK
IBRD • IDA | WORLD BANK GROUP

ReGenesees

Selected Tips and Tricks

(work in progress)

Diego Zardetto
World Bank STC for research

Software design principles (1/4)

- Formulas and computation techniques in the realm of Design-Based survey sampling involve both survey data and metadata
  - In particular, *proper* sampling variance estimation (for both Horvitz-Thompson and Calibration estimators) cannot be attained without carefully using *plenty* of metadata
- Typical examples of relevant metadata are information that describe:
  - the adopted sampling design (stages, strata, clusters, weights or inclusion probabilities, sampling fractions, …)
  - the adopted calibration procedure (assisting model, auxiliary variables, population totals, obtained g-weights, …)
- Thus, binding survey data and metadata in an effective and persistent way has to be considered a task of major importance when designing a good, multipurpose survey analysis software

Software design principles (2/4)

- Once survey data have been bound to sampling design and/or calibration metadata
  - All statistical analysis functions operating on sample data should automatically find and use the metadata appropriate to the current survey and the current analysis
- This principle is a fundamental pillar of ReGenesees
  - the user is first asked to simultaneously specify data and metadata so that both can be bound in a complex object
  - any subsequent analysis has then to be operated on that complex object by means of a dedicated statistical method
  - a user request of a statistical analysis (e.g. compute the estimate and the associated standard error of a given population parameter) on a given object is automatically dispatched to the right program, which depends on the class of the object

Software design principles (3/4)

- Less abstractly, consider the case of asking the software to estimate a give population parameter (for definiteness, say a Total), and to compute a confidence interval around the obtained estimate
- To do that, the software must compute an estimate of the sampling variance of the Total
- Now, the formula to be used for variance estimation critically depends on the sampling design, and is of course quite different for Horvitz-Thompson (HT) and Calibration (CAL) estimators
- Owing to the design principles illustrated so far, ReGenesees allows the user to exploit the same function in any case!

Software design principles (4/4)

- Indeed, driven by the metadata bound to the survey data, ReGenesees will automatically understand
  1. If HT or CAL estimators theory should be used
  2. How to take into account the sampling design
- Lastly, based on 1. and 2., ReGenesees will transparently call the right program
- To implement this behavior, an Object Oriented (OO) class system and method dispatching mechanism are used
- From an OO perspective, ReGenesees functions naturally fall into the following clear-cut decomposition:
  - Object (class instance) builders
  - Statistical analysis functions (main methods) operating on objects
  - Utility tools (ancillary methods)

An OO overview of ReGenesees (1/2)

- Object (class instance) builders
  - **`e.svydesign`** ⇨ class 'analytic' = [data + sampling design]
  - **`e.calibrate`** ⇨ class 'cal.analytic' = [data + sampling design + calibration info]
- Summary statistics functions (main methods)
  - **`svystatTM`** ⇨ Totals, Means, Frequencies
  - **`svystatR`** ⇨ Ratios
  - **`svystatS`** ⇨ Shares
  - **`svystatSR`** ⇨ Ratios of shares
  - **`svystatB`** ⇨ Regression coefficients
  - **`svystatQ`** ⇨ Quantiles
  - **`svystatL`** ⇨ Complex user-defined estimators

An OO overview of ReGenesees (2/2)

- For all the provided summary statistics, variance estimation methods are dispatched according to the class of the input design object:

  1. If the design object is non-calibrated (i.e. its class is 'analytic'), variance formulas are appropriate to Horvitz-Thompson estimators (and functions of them)

  2. If the design object is calibrated (i.e. its class is 'cal.analytic'), variance formulas are appropriate to Calibration estimators (and functions of them)

Model Formulae in ReGenesees

- All ReGenesees functions expect variables belonging to a design object be specified by means of R formulae
- This is deliberate: besides saving typing (compare `c('sex', 'age')` with `~ sex + age`) R formulae enable a terrific expressive power
- R formulae (whose traditional role is to define statistical models in a compact symbolic form, e.g. for `lm` and `glm` functions) are created by the tilde operator '~'
- Formulae are basically composed by terms separated by plus operators '+', which are used to add (not sum!) effects
- Each term is composed by (numeric or factor) variables names, perhaps separated by other operators
- Colon operators ':' are used to introduce interactions between variables in a term
- The minus operator '-' is used to remove a specified term from the formula. The statement '- 1' inside e formula removes the intercept
- The AsIs operator '`I()`' is used to "restore" the algebraic interpretation of the above operators