TEACHING SAMPLING TECHNIQUES USING R AND PYTHON

Jorge Luis Rueda[1]*, Beatriz Cobo[2], Luis Castro-Martín[3]
[1]Mr. Jorge Luis Rueda, University of Granada, SPAIN, jorgerueda279@correo.ugr.es
[2]Dr. Beatriz Cobo, University of Granada, SPAIN, beacr@ugr.es
[3]Dr. Luis Castro-Martín, Andalusian School of Public Health, SPAIN, luiscastro193@ugr.com
*Corresponding author

Abstract

The free software R (http://www.r-project.org) allows us to perform and obtain any type of statistical analysis needed, since it has available multiple function packages prepared to perform the task you want, in any field or area. This is thanks to its large community of users who are responsible for creating, developing, publishing and updating these packages. The field of sampling, which is key in any type of survey or research, has a wide range of functions and packages to be used, and in particular for complex sampling designs there are many packages that can help us to perform any technique we are interested in using. That is why in this paper we will address the use of the statistical environment of R for learning sampling for students of the degree in Statistics and any degree in which these contents are taught.

But it may be the case that the teachers are reluctant to use this programming software, despite being simple to use, simply because they are unfamiliar with it. Another programming software that is currently on the rise, and which has wide applications in fields as important for large companies as data mining, is the free software Python (https://www.python.org). More and more companies of all kinds are looking for employees with extensive knowledge in this software due to the infinite applications it has, being already one of the most used programming software in the world. In the case of statistical analysis, it has a large number of utilities, although in the specific case of sampling, especially for complex sample designs, there is not so much variety. In spite of this, the "samplics" package for Python, developed for selecting, weighting and analysing data from complex sampling designs, was released this November. Knowing about this package, in this paper we will also address the use of programming software Python for learning survey sampling techniques through this package, looking for similarities and differences between this method and the one we would use with programming software R.

Keywords: Teaching, Software Free, Sampling.

1 INTRODUCTION

Today we are surrounded by surveys of all kinds, some of which are crucial to our lives. Surveys on economic indicators can shape the course of a country or a city, as can surveys on social or health issues. This is why it is crucial to know if these surveys are carried out correctly, for which we need a basic statistical knowledge, as these surveys are usually obtained from a sample of a population. And if we want to go deeper, there are several quite complex sample designs, or techniques to refine estimates or to impute missing data, which require more advanced knowledge.

These more complex techniques are hardly applicable without the use of a computer, because of their difficulty and because they have been developed and widely applied thanks to the development of technology and programming software. This is why, no matter how much you want to teach these methods, if you want to apply them or practice them, you must make use of these software. There are many programming software, but the most interesting ones are the so-called free software, which can be obtained for free. Some of these software allow the users themselves to develop packages with different functions to carry out different operations in a specific field. This allows them to be used for almost any application, so they can be used for complex sampling methods. Together with the fact that they are freely available and that they offer an infinite variety of applications it is obvious that this software can be a very useful tool for teaching. In this paper we will focus on two free programming software, R (http://www.r-project.org) and Python (https://www.python.org), which offer an infinite number of applications and are widely used in the field of statistics (and many others).

For all these reasons, it is essential that if we teach these techniques in a practical way, we need a basic knowledge of these programmes in terms of what each one can offer us (in the form of function packages), synthesising the applications that they offer us. This is why in this work we are going to see how to use the statistical environment of R and Python for learning sampling for students of the degree in Statistics and any degree in which these contents are taught.

2 LEARNING SAMPLING TECHNIQUES WITH R

As we said in the introduction, in the free software R there are a multitude of function packages to perform almost any statistical operation. In this paper we will focus on the techniques for the analysis of a sample survey, talking about the packages that can be used for each one and giving a brief description of them.

2.1 Complex Sampling Designs: Estimates, Variances, and Calibrations

In this section we will see different function packages, from the R programming software, to obtain samples through different sampling designs (some of them more complex), and/or to calculate point estimates of the variables of interest and their variances.

- Package “Survey": Allows us to establish a sampling design for an extracted sample, with the aim of calculating point estimates and variances in a precise way. Among the sample designs we can set up are complex sample designs such as stratified sampling, cluster sampling, multistage sampling, and PPS (probability proportional to size) sampling with or without replacement.

The object we compute will be the one from which we will obtain estimates of totals, means, ratios and quantiles for domains or for entire populations, and also apply regression models. In the case of variance, we can estimate them from the estimator we have previously calculated, by linearisation or by resampling. Among these resampling techniques we highlight the BRR (balanced repeated replication) technique, the Jackknife method, and the Boostrap method. In the case of calibration, this package allows us to carry out post-stratification, generalised raking/calibration, and estimation for generalised regression.

- Package “Sampling”: It allows us to extract a sample using different complex sampling designs or algorithms such as: Brewer's algorithm, Midzuno's sampling, PPS sampling, Sampford's method, and balanced sampling. In terms of calibration, this package allows us to calibrate for non-response (with homogeneous response groups) in stratified samples, with the calib() function.

- Package “pps”: Allows to select samples using PPS sampling and using stratified simple random sampling. It can also calculate joint probabilities for PPS sampling, and for the Sampford’s method.

- Package “Stratification”: Allows to obtain the stratification of populations with a generalisation of the Lavallee-Hidiroglou’s method.

- Package “SamplingStrata”: Allows to obtain the best possible stratification of a sampling frame in a multivariate and multi-domain environment. In these strata the sampling sizes are determined in such a way that they meet the accuracy constraints on the target estimates. In addition, it can be used for the evaluation of the distributions of response variables in different strata.

- Package “Laeken”: It allows estimating some poverty indicators together with their variance for domains and strata using the Bootstrap method. Among these poverty indicators we highlight: poverty risk rate, quintile quota rate, median relative poverty risk gap, or the Gini coefficient. In terms of calibration, it allows us to do the same as the sampling package, but in this case the function is called calibWeights().

- Package “simFrame”: Allows comparison of estimates and user-defined variances in a simulation

environment.

- Package “lavaan.survey”: Allows for fitting structural equation models (SEM) on complex sample designs, allowing for the incorporation of clusters, strata, sampling weights, and finite population corrections in such SEM analyses.

- Package “Vardpoor”: Allows variance estimates to be computed using the linearisation technique and by the ultimate cluster method. We can also estimate variance for longitudinal and cross-sectional measure for any staged cluster sampling design.

- Package “reweight”: Allows calibration of weights for categorical survey data, resulting in marginal distributions that better match those of the population of interest, although it does not allow for complex sampling designs.

2.2 Imputation

Imputation is the substitution of values whose observations have been lost or not obtained (missing data) from other values. There are different estimation methods depending on how the estimation is performed, with packages for each of them. In the case of imputation methods based on the Expectation-Maximization (EM) algorithm, we highlight:

- Package "mi": allows multiple imputation by regression based on the iterative EM algorithm of missing values. The regression models can be user-defined, and the dataset can consist of continuous, semi-continuous, binary, and categorical variables.

- Package “mice”: Allows for multiple regression imputation based on iterative EM. The dataset can consist of continuous, binary and categorical variables.

For the case of imputation methods based on the nearest neighbour we highlight:

- Package "VIM": it allows implementing the sequential and randomised hot-deck algorithm, within a domain. It also provides a fast nearest-neighbour algorithm for large datasets.

- Package “yaImpute": Allows the implementation of imputation by means of nearest neighbour methods for continuous variables, being able to use different metrics and methods to determine the distance between observations.

2.3 Small Area Estimation

Small area estimation consists of estimating parameters on small subsets of an original population, whose sample size is small. For example, the case of small geographic territories within a larger territory. If the sampling design is designed to estimate parameters for the whole population and not for parts of it, the estimators will have the desired precision for the general population level, but not for the small area level. This is why we distinguish this estimation. For this type of estimation, we highlight the packages:

- Package "rsae": it allows estimating the parameters of the estimation model in small areas by the maximum likelihood method or by the robust maximum likelihood method. In addition, robust predictions can be calculated from these estimates.

- Package "nlme": Allows to fit linear Gaussian and non-linear mixed-effects models.

3 LEARNING SAMPLING TECHNIQUES WITH PYTHON: SAMPLICS PACKAGE

We have already seen how in R we have a multitude of packages to perform practically any operation related to sampling, especially on complex sampling designs. But there are other programming software that are widely used and are really interesting. One of them is the Python programming software, which also has several libraries in which you can perform all kinds of operations in any field. But for the analysis of complex survey samples there was no library available, which has now changed.

The “samplics” package was created by Mamadou Diallo in 2020 but has been constantly updated, adding more and more content on complex sample designs and different advanced techniques related to these designs. The last one was released this November, so everything we explain about this package will be related to what we have seen until this update. Possibly in the future it will have more applications, which would be fantastic.

This package allows us to draw samples using different simple sampling designs such as simple random sampling or systematic sampling, and more complex sampling designs such as stratified sampling, cluster

sampling, or PPS sampling. From these samples we can obtain estimates of the total, the mean, and the proportion, and other population parameters such as correlation coefficients, also allowing us to obtain estimates based on regression models. When estimating variances, we can do so by linearisation or by resampling, using methods such as the Bootstrap method, the Jackknife method, or the BBR technique. In addition, this package allows us to "update" the estimates by means of techniques such as calibration or post-stratification. Finally, this package also allows us to make estimates in small areas.

4 CONCLUDING REMARKS

In this work we have made a summary of all the packages, both R and Python programming software, related to sampling techniques, especially the more advanced ones. This gives us a broad overview of how these techniques could be taught to students of the Statistics degree or degrees in which this type of content is taught, using these programmes. It remains for everyone to explore beyond each package and to differentiate the large number of functions contained in each package explained.

REFERENCE LIST

Lumley, T. Survey: Analysis of complex survey samples., R package version 3.30-3. http://cran.r-project.org/web/packages/survey/index.html

Templ, M. CRAN Task View. Official Statistics & Survey Methodology., R package version 2015-04-12. http://cran.rproject.org/web/views/OfficialStatistics.html

Tillé, Y., Matei, A. Sampling: Survey Sampling., R package version 2.6. http://cran.r-project.org/web/packages/sampling/index.html

Diallo, M. S. (2021). Samplics: A Python Package for selecting, weighting and analyzing data from complex sampling designs. *Journal of Open Source Software*, *6*(68), 3376.