Nuevos métodos para la enseñanza de muestreo con R: el paquete 'TeachingSampling'

Arcos, Antonio, Rueda, María del Mar, Molina, David

Universidad de Granada

Resumen

Que la forma de enseñar ha experimentado cambios muy importantes en los últimos años es un hecho evidente. Los continuos avances en materia científica y, sobre todo, el vertiginoso progreso de la tecnología han motivado una revisión de los métodos de enseñanza y las técnicas didácticas tradicionales que ha desembocado en la actualización de todos ellos con el fin de adaptarlos a las necesidades presentes.

Esta renovación metodológica, que ha afectado a todas las áreas de conocimiento sin excepción, puede constatarse, por ejemplo, en la reciente inclusión en el software científico de módulos específicos para la enseñanza de contenidos. El paquete TeachingSampling, incluido dentro de la librería de paquetes del programa estadístico R, constituye un buen ejemplo de ello.

En este trabajo se desarrollará un análisis detallado de las principales funciones del paquete TeachingSampling, el cual se acompañará de ejemplos prácticos para ilustrar su funcionamiento.

Palabras clave: muestreo en poblaciones finitas, R, TeachingSampling.

1. Introducción

La forma de enseñar ha cambiado. Para comprobarlo, basta con volver la vista atrás unos años y comparar los métodos y materiales que se empleaban entonces con los que se utilizan ahora. La irrupción de la tecnología en las aulas ha sido la principal responsable de que las técnicas docentes que se venían utilizando hasta hace unos pocos años hayan quedado obsoletas. Si a esto le sumamos que la sociedad actual demanda profesionales cada vez más formados en aspectos prácticos y que sepan enfrentarse a problemas reales más allá de los teóricos, la renovación metodológica que la enseñanza viene experimentando de un tiempo a esta parte no solo queda sobradamente justificada, sino que se hace necesaria.

Estos cambios han quedado puestos de manifiesto en todos los niveles educativos. Ya en la educación primaria es cada vez más habitual que los alumnos dispongan de ordenadores o tablets para ayudarles en su aprendizaje, lo cual, hasta hace poco, era casi impensable. Por otra parte, a nivel universitario, las tradicionales lecciones magistrales en el aula se complementan ahora con seminarios prácticos que completan la formación del alumno, enseñándole el manejo de software propio de su especialidad.

Por todo ello, ha proliferado una gran cantidad de software informático para el apoyo a la docencia y, para el software ya existente, se han desarrollado nuevas funcionalidades para el mismo fin. Dentro de este segundo grupo se encuadra R. R es un programa de carácter libre para el análisis estadístico de datos. Pero es algo más que eso: R constituye un poderoso entorno de programación que permite al usuario la creación de nuevas funciones que satisfagan sus necesidades específicas ante un determinado problema. Precisamente esta flexibilidad junto a su gratuidad han hecho que la popularidad de R aumente enormemente entre la comunidad científica, convirtiéndolo en uno de los programas estadísticos más utilizados, en detrimento de la mayoría del software comercial.

En J. M. Contreras, G. R. Cañadas, M. M. Gea y P. Arteaga (Eds.), Actas de las Jornadas Virtuales en Didáctica de la Estadística, Probabilidad y Combinatoria (pp. 433-438). Granada, Departamento de Didáctica de la Matemática de la Universidad de Granada, 2013.

Otra de las ventajas que R proporciona a sus usuarios es la posibilidad de extender su configuración básica mediante la instalación de paquetes. Un paquete no es más que un conjunto de funciones, desarrolladas habitualmente por investigadores y especialistas en el ámbito de la estadística, relativas a una temática común. Actualmente, existen un total de 4142 paquetes disponibles para su instalación. Una pequeña parte de ellos, los denominados paquetes básicos, forman parte de la instalación inicial del programa, mientras que el resto son de instalación opcional. Los paquetes ofrecen una gama de funcionalidades muy amplia: desde análisis estadísticos complejos hasta herramientas para la elaboración de gráficos. También hay otros paquetes que proporcionan herramientas para la enseñanza y el autoaprendizaje de contenidos, como, por ejemplo, 'TeachingSampling'. En la siguiente sección analizaremos en profundidad las funciones de este paquete.

2. El paquete 'TeachingSampling'

El paquete 'TeachingSampling' de R (Gutiérrez Rojas, 2009) es un conjunto de herramientas para el muestreo y la estimación de parámetros. Según palabras del propio autor, "el paquete 'TeachingSampling' fue pensado como protagonista en la enseñanza del muestreo y en el aula de clase".

La primera versión del paquete (0.7.6) se lanzó en marzo de 2009. Durante los siguientes años vieron la luz nuevas versiones que corregían errores e incluían funciones adicionales; hasta llegar a la séptima versión (2.0.1), lanzada en abril de 2011, la cual es la más reciente versión de 'TeachingSampling' de que se dispone a día de hoy. En la tabla 1 se muestra cada una de las versiones del paquete junto con su fecha de lanzamiento.

Tabla 1. Versiones de 'TeachingSampling'

	Fecha de lanzamiento
0.7.6	5 de marzo de 2009
0.8.1	24 de mayo de 2009
1.0.2	3 de septiembre de 2009
1.1.9	18 de enero de 2010
1.4.9	11 de marzo de 2010
1.7.9	24 de julio de 2010
2.0.1	1 de abril de 2011

La última versión de 'TeachingSampling' cuenta con un total de 45 funciones. Un buen número de ellas tiene por objeto la extracción de muestras bajo diversos diseños muestrales. Así, por ejemplo, mediante las funciones *S.WR* y *S.SI* se puede obtener una muestra aleatoria simple con y sin reemplazamiento, respectivamente.

Igualmente, se pueden seleccionar muestras a partir de diseños muestrales más complejos, como el muestreo sistemático (mediante la función *S.SY*), el muestreo de Poisson (por medio de la función *S.PO*) o el muestreo con probabilidades proporcionales al tamaño, ya sea con o sin reposición (a través de la funciones *S.PPS* y *S.piPS*, respectivamente), por mencionar algunos. Una vez que la muestra ha sido seleccionada, 'TeachingSampling' hace posible la estimación de parámetros ya que, para cada técnica de muestreo, dispone de una función que calcula una estimación del total poblacional, su varianza estimada y su coeficiente de variación estimado. Por tanto, es muy habitual en la práctica que las funciones de selección de muestras y de estimación de parámetros se utilicen secuencialmente, tal y como veremos en el ejemplo que propondremos en la siguiente sección.

El paquete incluye, además, algunas funciones elementales relativas a la teoría del muestreo en poblaciones finitas, entre las que se encuentran *HH*, para el cálculo del estimador de Hansen-Hurwitz (1943) y *HT* y *varHT*, que permiten la obtención del estimador de Horvitz-Thompson (1952) y su varianza. Del mismo modo, a partir de las probabilidades de selección, del tamaño de la población y del tamaño de la muestra se

pueden calcular las probabilidades de inclusión de primer y segundo orden para diseños muestrales de tamaño fijo sin reemplazamiento mediante el empleo de las funciones *Pik* y *Pikl*. Cuando el diseño muestral de tamaño fijo es un diseño con probabilidades proporcionales al tamaño, la función *PikPPS* posibilita el cálculo de las probabilidades de inclusión de primer orden. Una última función interesante que 'TeachinSampling' incorpora en relación a las probabilidades de inclusión es *PikHol*, mediante la cual se pueden obtener las probabilidades de inclusión óptimas de primer orden según la aproximación de Holmberg (2002).

Otro de los puntos fuertes del paquete son sus herramientas para la elaboración de diseños muestrales, con o sin reemplazamiento. Así, dada una población cualquiera de individuos, la función *SupportRS* permite calcular el conjunto de todas las posibles muestras sin reemplazamiento de cualquier tamaño, esto es, su soporte. Si se especifica un tamaño muestral concreto, se pueden emplear las funciones *SupportWR* o *Support* para obtener todas las muestras con o sin reemplazamiento de ese tamaño que se pueden extraer de la población de individuos considerada. Las funciones *IkRS*, *IkWR* e *Ik* complementan a las anteriores, proporcionando como salida sendas matrices binarias, con elementos a_{ij}, en las que las i filas recogen cada una de las posibles muestras dada una población, mientras que las j columnas representan los elementos de dicha población. En aquellos casos en que el j-ésimo individuo pertenezca a la i-ésima muestra, el elemento a_{ij} tomará el valor 1, siendo este valor 0 en cualquier otro caso. Como ya ocurrió con las funciones para el cálculo del soporte, el uso de una u otra función dependerá del tipo de diseño que se considere. Para diseños sin reemplazamiento en los que no se especifica el número de elementos de la muestra, se usa la función *IkRS*. Si, por el contrario, se cuenta con un tamaño muestral concreto, la función a usar es *Ik* cuando el diseño no permite el reemplazo de las unidades o *IkWR* en caso contrario.

Para diseños con reemplazo, 'TeachingSampling' incluye algunas funciones adicionales, como son *OrderWR* y *nk*. La primera de ellas permite obtener el conjunto de todas las muestras, dado el tamaño de las mismas, cuando se considera un diseño muestral ordenado con reemplazamiento. Nótese que todas las funciones descritas hasta ahora no tenían en cuenta el orden de las muestras seleccionadas. Por su parte, mediante *nk* podemos determinar el número de veces que cada elemento de una población aparece en cada posible muestra en un diseño no ordenado con reemplazo.

El paquete también incluye dos conjuntos de datos propios, bautizados por el autor como *Lucy* y *Marco*, los cuales se usan en los ejemplos que se incluyen en el manual de uso del paquete para mostrar el funcionamiento de la mayoría de las funciones que lo integran.

En la siguiente sección, se darán unas nociones básicas para la instalación del paquete y se aplicarán algunas de las funciones consideradas a lo largo de esta sección a un conjunto de datos reales.

3. Una aplicación práctica

Para instalar el paquete mientras se está ejecutando una sesión de R, se puede usar el comando *install.packages ("TeachingSampling").* Para cargarlo, se emplea este otro comando: *library("TeachingSampling")*. Al ejecutar el comando *help(package = TeachinSampling)*, se obtiene una lista con todas las funciones del paquete. Si se necesita información adicional sobre alguna función en particular, se puede utilizar el comando *help(nombre)*, sustituyendo *nombre* por el nombre concreto de la función a consultar.

Para mostrar cómo funcionan algunas de las utilidades de 'TeachingSampling', se va a considerar el conjunto de datos formado por las siguientes variables:

- *provincia*. Nombre de cada una de las 52 provincias españolas, incluyendo las ciudades autónomas de Ceuta y Melilla.

- *poblacion*. Población total.
- *pib*. Producto interior bruto per cápita, en euros.

Se supondrá que el conjunto de datos ha sido leído correctamente y se ha almacenado en la variable *datos*.

```
datos <– read.table ("provincias.txt", header = T)
```

Comenzaremos seleccionando, por ejemplo, una muestra aleatoria simple sin reemplazamiento de tamaño 8 para estimar la media del producto interior bruto español a partir de la información que nos proporcione.

```
muestra <– S.SI (52, 8)
```

Se puede consultar qué elementos han sido seleccionados en la muestra mediante la siguiente orden

```
datos_muestra <– datos[muestra,]
```

El siguiente paso consiste es utilizar la función *E.SI* para calcular la estimación del total de la variable de interés.

```
E.SI(52,8,datos_muestra$pib)
```

Los resultados de la ejecución de la orden anterior en nuestro caso pueden consultarse en la figura 1.

```
                              y
Estimation 1.119645e+06
Variance   8.947985e+09
CVE        8.448556e+00
```

Figura 1. Estimación mediante muestreo aleatorio simple sin reemplazamiento

Puesto que la estimación que buscamos es la de la media de la variable y no la de su total, habrá que ajustar los resultados que se obtengan de la ejecución de la función anterior, teniendo en cuenta las relaciones entre las estimaciones del total y de la media de una variable, para así conseguir los valores deseados.

Nos proponemos ahora la estimación de la misma cantidad pero considerando un muestreo sin reemplazo con probabilidades proporcionales al tamaño en el que la muestra de provincias se va a seleccionar teniendo en cuenta la población de cada una de ellas. Así, las provincias con mayor número de habitantes tendrán una probabilidad mayor de formar parte de la muestra que aquellas otras menos habitadas.

```
muestra <– S.piPS (8, datos$poblacion)
```

Aprovechamos que la función *S.piPS* devuelve tanto las unidades seleccionadas en la muestras como sus probabilidades de inclusión y las almacenamos en variables diferentes: ambas nos serán necesarias a lo largo del proceso de estimación.

```
unidades_seleccionadas <– muestra[,1]
probabilidades_seleccion <– muestra[,2]
```

La información referente a las unidades seleccionadas se puede consultar ejecutando el siguiente comando

```
datos_muestra <– datos[unidades_seleccionadas,]
```

A continuación, estimamos el total de la variable de interés mediante la función *E.piPS*. Para ello, necesitaremos hacer uso de las probabilidades de inclusión previamente calculadas.

E.piPS (datos_muestra$pib, probabilidades_seleccion)

```
                              y
Estimation 7.150583e+05
Variance   1.583873e+10
CVE        1.760025e+01
```

Figura 2. Estimación mediante muestreo con probabilidades proporcionales al tamaño sin reemplazamiento

Los resultados que se obtienen son los que se muestran en la figura 2. Ya solo queda ajustar los resultados de forma adecuada para llegar a la estimación buscada, tal y como se ha comentado anteriormente.

Restrinjámonos ahora al conjunto de datos relativos a las provincias gallegas únicamente.

datos_galicia <– read.table ("provincias_galicia.txt", header = T)

Supongamos que nos gustaría extraer todas las posibles muestras sin reemplazamiento que se pueden obtener de esta población, sea cual sea el tamaño de las mismas. Para ello, habríamos de emplear la siguiente orden

SupportRS (4, ID = datos_galicia$provincia)

La salida que se obtendría sería la que se muestra en la figura 3. En ella vemos que, efectivamente, se consideran desde la muestra vacía hasta la muestra que incluye todos los elementos de la población, esto es, el censo.

```
      [,1]          [,2]          [,3]          [,4]
 [1,] NA            NA            NA            NA
 [2,] "ACoruña"     NA            NA            NA
 [3,] "Lugo"        NA            NA            NA
 [4,] "Ourense"     NA            NA            NA
 [5,] "Pontevedra"  NA            NA            NA
 [6,] "ACoruña"     "Lugo"        NA            NA
 [7,] "ACoruña"     "Ourense"     NA            NA
 [8,] "ACoruña"     "Pontevedra"  NA            NA
 [9,] "Lugo"        "Ourense"     NA            NA
[10,] "Lugo"        "Pontevedra"  NA            NA
[11,] "Ourense"     "Pontevedra"  NA            NA
[12,] "ACoruña"     "Lugo"        "Ourense"     NA
[13,] "ACoruña"     "Lugo"        "Pontevedra"  NA
[14,] "ACoruña"     "Ourense"     "Pontevedra"  NA
[15,] "Lugo"        "Ourense"     "Pontevedra"  NA
[16,] "ACoruña"     "Lugo"        "Ourense"     "Pontevedra"
```

Figura 3. Conjunto de todas las muestras sin reemplazamiento para los datos de las provincias de Galicia

Si el reemplazamiento estuviera permitido, y se quisiera seleccionar una muestra de tamaño 2, ¿cuáles serían las opciones posibles? Para responder a esta pregunta, ejecutaríamos el siguiente comando:

SupportWR (4, 2, ID = datos_galicia$provincia)

el cual proporcionaría como salida las 10 muestras que se recogen en la figura 4.

```
      [,1]         [,2]
 [1,] "ACoruña"    "ACoruña"
 [2,] "ACoruña"    "Lugo"
 [3,] "ACoruña"    "Ourense"
 [4,] "ACoruña"    "Pontevedra"
 [5,] "Lugo"       "Lugo"
 [6,] "Lugo"       "Ourense"
 [7,] "Lugo"       "Pontevedra"
 [8,] "Ourense"    "Ourense"
 [9,] "Ourense"    "Pontevedra"
[10,] "Pontevedra" "Pontevedra"
```

Figura 4. Conjunto de todas las muestras de tamaño 2 con reemplazamiento para el conjunto de datos de las provincias de Galicia

4. Conclusiones

El paquete 'TeachingSampling' de R constituye una herramienta que permite a sus usuarios un primer acercamiento a la teoría del muestreo en poblaciones finitas. Puesto que incluye una gran parte de las funciones básicas relativas al muestreo en poblaciones finitas, su uso resulta idóneo en cursos o asignaturas de carácter introductorio sobre esta materia, en los que el docente podría emplearlo, por ejemplo, para ilustrar la teoría expuesta en el aula. Además, dada su sencillez, también se puede utilizar en el aprendizaje autónomo.

En caso de que fuera necesario el empleo de diseños muestrales más complejos o la utilización de herramientas para la calibración o para el tratamiento de la no respuesta, 'TeachingSampling' puede complementarse con las funciones del paquete 'sampling' (Matei y Tillé, 2009)

Referencias

Gutiérrez Rojas, A. (2009). Manual de uso del paquete 'TeachingSampling'. Extraído el 22 de marzo de 2013, de http://cran.r-project.org/web/packages/TeachingSampling/TeachingSampling.pdf

Hansen, M. H. y Hurwitz, W. N. (1943). "On the theory of sampling from finite populations". *Annals of Mathematical Statistics, 14*: 333-362.

Holmberg, A. (2002). "On the choice of sampling design under GREG estimation in multiparameter surveys", *R&D Report 2002:1, Statistic Sweden*

Horvitz,D. G. y Thompson, D. J. (1952). "A generalization of sampling without replacement from a finite universe", *Journal of the American Statistical Association*, 47: 663–685

Matei, A. y Tillé, Y. (2009). Manual de uso del paquete 'sampling'. Extraído el 22 de marzo de 2013, de http://cran.r-project.org/web/packages/sampling/sampling.pdf