Lección 1

Conceptos básicos de muestreo estadístico

En todo estudio estadístico hemos de distinguir entre *población*, que es un conjunto de individuos con una característica observable, y *muestra*, un subconjunto de una población del que se espera que la represente. Por ejemplo, los estudiantes de grado de la UIB serían una población, y si escogiéramos al azar un estudiante de cada grado, obtendríamos una muestra. Pero también podríamos considerar los estudiantes de grado de la UIB como una muestra de la población de los estudiantes universitarios españoles: depende del estudio que queramos realizar.

En la lección 15 del primer volumen distinguíamos dos tipos de análisis de un conjunto de datos sobre un determinado fenómeno. En aquel momento nos centrábamos en el *análisis exploratorio*, en el que describimos, resumimos e intentamos interpretar los datos usando técnicas de estadística descriptiva. Pero en otro tipo de estudios, nuestro objetivo es contrastar una hipótesis sobre el comportamiento de dicho fenómeno. Para hacerlo, tendremos que llevar a cabo un estudio estadístico que, al menos en principio, nos permita confirmar o refutar dicha hipótesis. Este tipo de estudio estadístico pertenece al ámbito del *análisis de datos confirmatorio*, ya que persigue reforzar y aportar evidencias sobre la veracidad de nuestra hipótesis. En el análisis de datos de tipo confirmatorio, se usan técnicas de *estadística inferencial*, cuyo objetivo es *inferir información* sobre el conjunto de una población, es decir, contestar preguntas sobre el total de la población, a partir del estudio de una muestra representativa.

Los pasos siguientes son los habituales en un estudio confirmatorio:

(1) Establecer la hipótesis que se desea contrastar.

(2) Determinar la información que se necesita para hacerlo.

(3) Diseñar un experimento; este paso incluye:

- Seleccionar los individuos de la muestra.
- Elegir las técnicas inferenciales adecuadas.

(4) Obtener la información de los individuos de la muestra.

(5) Aplicar las técnicas de inferencia elegidas con el *software* adecuado.

(6) Obtener conclusiones.

(7) Si las conclusiones son fiables y suficientes, redactar un informe; en caso contrario, volver a empezar.

En esta lección nos concentramos en los diferentes tipos de *técnicas de muestreo*: los métodos generales que permiten seleccionar muestras representativas de la población y de esta manera llevar a cabo el tercer paso de la lista anterior. Existen muchas técnicas de muestreo, cada una de las cuales proporciona una muestra representativa de la población. A continuación describimos de forma breve algunos tipos.

Muestreo aleatorio con y sin reposición. Un *muestreo aleatorio* consiste en seleccionar una muestra de la población de manera que todas las muestras del mismo tamaño tengan la misma probabilidad; es decir, que si fijamos el número de individuos de la muestra, cualquier conjunto de ese número de individuos ha de tener la misma probabilidad de ser seleccionado. Hay dos tipos básicos de muestreos aleatorios que vale la pena distinguir.

Pensemos en una urna con 100 bolas de colores. Una manera de obtener una muestra de 10 bolas sería repetir 10 veces el proceso de sacar una bola de la urna, observar su color y devolverla a la urna. Este tipo de muestreo recibe el nombre de *muestreo con reposición* o *muestreo aleatorio simple*. Observad que en este tipo de muestreo un mismo individuo puede aparecer varias veces en una muestra, y que todas las muestras de 10 individuos "con posibles repeticiones" tienen la misma probabilidad.

Otra manera de obtener una muestra de nuestra urna sería repetir 10 veces el proceso de sacar una bola de la urna pero ahora no devolverla. Esto es equivalente a extraer de golpe 10 bolas de la urna. Estas muestras no tienen individuos repetidos, y cualquier selección de 10 bolas diferentes se puede obtener con la misma probabilidad. En este caso se habla de *muestreo aleatorio sin reposición*.

Cuando el tamaño de la población es muy grande en relación a la muestra y por lo tanto la probabilidad de que un individuo se repita en la muestra es muy pequeña, el muestreo aleatorio con reposición y el muestreo aleatorio sin reposición son aproximadamente equivalentes. Por lo tanto, la mayoría de fórmulas que daremos en este curso suponiendo que trabajamos con una muestra aleatoria simple las consideraremos igualmente válidas para muestras aleatorias sin reposición, siempre y cuando el tamaño de la población sea muy grande en relación al de la muestra. Si el tamaño de la población es relativamente pequeño, algunas de estas fórmulas se pueden salvar aplicando correcciones adecuadas para compensar el efecto del tamaño de la población.

Muestreo aleatorio estratificado. Este tipo de muestreos se utiliza cuando la población está dividida en grupos o estratos y estos son de interés para la variable estudiada. En este

caso, se toman muestras cuya composición por estratos mantenga las proporciones de la población original; es decir, el tamaño de la muestra de cada estrato ha de representar el mismo porcentaje del total de la muestra que el estrato correspondiente en la población completa. Una vez determinados de esta manera los tamaños de las muestras de los diferentes estratos de la población, se obtiene una muestra de cada uno de ellos mediante un muestreo aleatorio con o sin reposición.

Por ejemplo los estratos podrían ser los grupos de edad, y entonces la muestra en cada grupo de edad se tomaría proporcional a la fracción que representa dicho grupo de edad en la población total. O, en las Islas Baleares, los estratos podrían ser las islas, y la muestra tomada en cada isla sería proporcional a la población relativa de la misma dentro del conjunto total de la comunidad autónoma. O, en una provincia, los estratos podrían ser los municipios, o podrían distinguir el nivel educativo de sus habitantes, etc. Depende de la propiedad a estudiar y de qué estratos se consideran relevantes para dicha propiedad.

Muestreo por conglomerados. El proceso de obtener una muestra aleatoria en algunos casos es caro o difícil. Por ejemplo, si el estudio se realiza sobre conjuntos de personas, tener una lista completa de dichas personas puede ser muy costoso. Imaginemos que queremos estudiar los hábitos de alimentación que tienen los estudiantes de Primaria de Baleares. Para ello, previo permiso de la autoridad competente, tendremos que seleccionar una muestra representativa de los escolares de Baleares. Pero, en vez de extraer una muestra representativa de todos los estudiantes de Primaria, lo que haríamos sería primero escoger al azar un conjunto de colegios, a los que llamamos en este contexto *conglomerados*, y a continuación, dentro de cada colegio (conglomerado) elegiríamos al azar un conjunto de estudiantes. Pensemos que es mucho más sencillo poseer la lista completa de estudiantes de unos pocos colegios que conseguir la lista completa de todos los estudiantes de todos los colegios, y que es más barato ir a unos pocos colegios concretos que ir a todos los colegios de las Islas a entrevistar a unos pocos estudiantes en cada centro. De manera similar, algunos estudios poblacionales a nivel estatal se realizan solamente en algunas provincias escogidas aleatoriamente.

Muestreos no aleatorios. Cuando la selección de la muestra no es aleatoria, se habla de *muestreo no aleatorio*. Es el tipo más frecuente de muestreo porque, en muchos casos, nos tenemos que conformar con la información disponible o la obtenida voluntariamente. Por ejemplo, en la UIB, para estudiar la opinión que de un profesor tienen los alumnos de una clase, se considera sólo la muestra de los estudiantes que voluntariamente han rellenado la encuesta de opinión, una muestra que de ninguna manera es aleatoria: el perfil del estudiante que contesta voluntariamente una encuesta de este tipo está muy definido. En este caso se trataría de una *muestra autoseleccionada*.

Otro tipo de muestreos no aleatorios son los *oportunistas*. Se da este caso, por ejemplo, cuando se realiza una encuesta telefónica: las personas que disponen de teléfono fijo no tienen por qué ser representativas de la sociedad para el tema concreto de la encuesta. O por ejemplo, supongamos que queremos estudiar una característica de los animales de una determinada especie en un hábitat, y la medimos en los animales que capturamos u observamos. Estos ejemplares no tienen por qué ser representativos de la población: por

ejemplo, a lo mejor son los menos espabilados. O imaginad que tenéis una bolsa con bolas de diferentes tamaños. Si las removéis bien, las pequeñas tenderán a ir a parar al fondo y las grandes a quedar en la parte superior. Por lo tanto, si tomáis una muestra de la capa superior (que será lo más cómodo), no será representativa del total de la bolsa.

Existen otros tipos de muestreo que suelen ser combinaciones de las técnicas anteriores u otros tipos de técnicas. En cualquier caso, lo importante es recordar que el estudio estadístico que se realice *a posteriori* deberá ser diferente según el muestreo realizado.

Una vez realizado el muestreo y obtenidos los datos (los llamados *datos brutos*, *raw data*, de los que hablábamos en la lección sobre datos cuantitativos agrupados en el primer volumen), el siguiente paso es inferir información a partir de dichos datos. Como ya hemos indicado anteriormente, esto significa intentar obtener información de la población a partir de dichos datos. Dicha información puede obtenerse de dos formas:

(1) Suponiendo que conocemos el modelo al que se ajusta la población: es decir, suponiendo que conocemos el tipo de distribución de la variable aleatoria que modela el objeto de estudio de la población pero desconocemos uno o varios parámetros de los que depende dicha distribución. Por ejemplo, podemos saber que la altura de los habitantes de un municipio es una variable aleatoria normal, pero desconocer su parámetro μ (media) o su parámetro σ (desviación típica), o ambos. Si estamos en este caso, hablaremos de *estimación paramétrica*.

(2) Suponiendo que desconocemos el modelo o la variable aleatoria que modela el objeto de estudio de la población. En este caso, hablaremos de *estimación no paramétrica*.

En el caso de la estadística paramétrica, existen tres vías para obtener información sobre los parámetros:

- *Estimación puntual*. Se trata de obtener fórmulas, llamadas *estadísticos* o *estimadores*, que se aplican a los datos de la muestra para obtener una aproximación (el término exacto es una *estimación*) del valor de dicho parámetro para la población. Por ejemplo, la media aritmética de los datos $x_1, \ldots, x_n$ de una muestra,

$$\overline{x} = \frac{x_1 + \cdots + x_n}{n},$$

  es un estimador del valor medio, o *esperanza*, de una variable aleatoria. Naturalmente, tendremos que demostrar algunos teoremas que nos digan cuándo y hasta qué punto esta estimación es fiable.

- *Intervalos de confianza*. Se trata de obtener intervalos que contengan con probabilidad alta el parámetro objeto de estudio. Trataremos esta parte en la próxima lección.

- *Contraste de hipótesis*. Se establecen dos hipótesis sobre el parámetro y se contrastan. Los estudiaremos en próximas lecciones.

En este curso estudiaremos técnicas de estimación solamente para el caso de *muestreo aleatorio simple*, es decir, al azar y con reposición, o al azar sin reposición si la población es muy grande. Recordemos que un método de selección al azar de muestras de tamaño n (es decir, formadas por n individuos) de una población de tamaño N produce *muestras aleatorias simples* (*m.a.s.*, para abreviar) cuando, siempre que lo aplicamos a una misma población, todas las muestras posibles de n individuos (con posibles repeticiones) tienen la misma probabilidad de ser elegidas. El tener una m.a.s. de una población junto con un tamaño muestral adecuado n nos asegurará que la muestra sea representativa.

La manera más sencilla de llevar a cabo un muestreo aleatorio simple es numerar todos los individuos de una población y sortearlos eligiendo números de uno en uno como si se tratase de una lotería, por ejemplo con algún generador de números aleatorios. Esto se puede llevar a cabo fácilmente con `R`.

`R` dispone de un generador de muestras aleatorias de un vector. La función básica es

```
sample(x, n, replace=...)
```

donde:

- `x` es un vector o un número natural x, en cuyo caso `R` entiende que representa el vector $1, 2, \ldots, x$;
- `n` es el tamaño de la muestra que deseamos extraer;
- el parámetro `replace` puede igualarse a `TRUE`, y será una muestra aleatoria con reposición, es decir, simple, o a `FALSE`, y será una muestra aleatoria sin reposición. Este último es su valor por defecto, por lo que no es necesario especificarlo.

Los dos primeros parámetros han de entrarse en este orden.

Por lo tanto, por ejemplo, para obtener una m.a.s. de 15 números entre 1 y 100, podemos ejecutar:

```
> sample(100,15,replace=TRUE)
 [1]  66 100  72   3  21  66  36  51 100  29  79  15  44  70  81
```

Naturalmente, cada ejecución de `sample` con los mismos parámetros puede dar lugar a muestras diferentes, y todas ellas tienen la misma probabilidad de aparecer:

```
> sample(100,15,replace=TRUE)
 [1] 67 16 29 69 77 63 62 16 67 73 56  6 30 99 14
> sample(100,15,replace=TRUE)
 [1] 44 69 85  6 22 34  8 19 77 14 81 31 65 22 92
> sample(100,15,replace=TRUE)
 [1] 67 17  5 61 57  9 96 70 82 53 85 53 72 55 69
```

A modo de ejemplo, recordemos el *data frame* `iris`, que recoge medidas de pétalos y sépalos de 150 flores de tres especies de iris.

```
> str(iris)
'data.frame': 150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5
   ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1
   ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1
   1 1 1 1 1 1 1 1 1 ...
```

Si queremos extraer una m.a.s. de 15 ejemplares (filas) de esta tabla de datos, podemos hacer lo siguiente:

```
> x=sample(dim(iris)[1],15,replace=TRUE) #Índices de la m.a.s.
> muestra_iris=iris[x,] #La m.a.s. de la tabla iris
> muestra_iris
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
77           6.8         2.8          4.8         1.4 versicolor
89           5.6         3.0          4.1         1.3 versicolor
15           5.8         4.0          1.2         0.2     setosa
124          6.3         2.7          4.9         1.8  virginica
100          5.7         2.8          4.1         1.3 versicolor
148          6.5         3.0          5.2         2.0  virginica
79           6.0         2.9          4.5         1.5 versicolor
56           5.7         2.8          4.5         1.3 versicolor
138          6.4         3.1          5.5         1.8  virginica
71           5.9         3.2          4.8         1.8 versicolor
144          6.8         3.2          5.9         2.3  virginica
70           5.6         2.5          3.9         1.1 versicolor
21           5.4         3.4          1.7         0.2     setosa
132          7.9         3.8          6.4         2.0  virginica
36           5.0         3.2          1.2         0.2     setosa
```

Recordad que `dim` aplicado a un *dataframe* nos da un vector con sus dimensiones: su número de filas y su número de columnas, en este orden. Por lo tanto, `dim(iris)[1]` es el número de filas de `iris`.

Si quisiéramos una muestra aleatoria de longitudes de pétalos, podríamos aplicar directamente `sample` al vector correspondiente:

```
> muestra_long_pet=sample(iris$Petal.Length,15,replace=TRUE)
> muestra_long_pet
 [1] 1.4 4.8 4.9 4.9 4.5 3.7 1.6 3.3 5.6 1.3 3.9 5.1 6.0 1.6 4.0
```

El hecho de que funciones como `sample` o los generadores de vectores numéricos con una cierta distribución de probabilidad fijada, como `rnorm` o `rbinom`, produzcan... pues eso, vectores aleatorios, puede tener inconvenientes a la hora de reproducir una simulación.

`R` permite «fijar» el resultado de una función aleatoria con la instrucción `set.seed`. Sin entrar en detalles sobre cómo funcionan, los diferentes algoritmos que usa `R` para generar números aleatorios usan una *semilla*, que se modifica después de la ejecución del algoritmo, y por eso cada vez dan un resultado distinto. Pero, para una semilla fija, el algoritmo da el mismo resultado siempre. Lo que hace la función `set.seed` es igualar esta semilla al valor que le entramos. Si tras aplicar esta función a un número concreto ejecutamos una instrucción que genere un vector aleatorio de una longitud fija con una distribución fija, el resultado será siempre el mismo. Veamos un ejemplo:

```
> rnorm(5)
[1]  0.7891883 -0.1660983  0.8558162 -1.0170676  0.4872311
> set.seed(20)
> rnorm(5)
[1]  1.1626853 -0.5859245  1.7854650 -1.3325937 -0.4465668
> set.seed(20)
> rnorm(5)
[1]  1.1626853 -0.5859245  1.7854650 -1.3325937 -0.4465668
> rnorm(5)
[1]  0.5696061 -2.8897176 -0.8690183 -0.4617027 -0.5555409
> set.seed(10)
> rnorm(5)
[1]  0.01874617 -0.18425254 -1.37133055 -0.59916772  0.29454513
> set.seed(10)
> rnorm(5)
[1]  0.01874617 -0.18425254 -1.37133055 -0.59916772  0.29454513
```

Ejecutado inmediatamente después de `set.seed(20)`, `rnorm(5)` siempre da lo mismo. Y ejecutado después de `set.seed(10)`, `rnorm(5)` vuelve a dar siempre da lo mismo, pero diferente de con `set.seed(20)`.

La función `set.seed` no sólo fija el resultado de la siguiente instrucción que genere un vector aleatorio, sino que, como fija la semilla de aleatoriedad y las funciones siguientes la modifican de manera determinista, también fija los resultados de todas las instrucciones siguientes que generen vectores aleatorios.

```
> set.seed(100)
> sample(10,3)
[1] 4 3 5
> sample(10,3)
[1] 1 5 4
> sample(10,3)
[1] 9 4 5
> set.seed(100)
> sample(10,3)
[1] 4 3 5
> sample(10,3)
[1] 1 5 4
```

```
> sample(10,3)
[1] 9 4 5
```

Si queréis volver a «reiniciar» la semilla de la aleatoriedad tras haber usado un `set.seed`, podéis usar `set.seed(NULL)`.

```
> set.seed(100)
> sample(10,3)
[1] 4 3 5
> set.seed(NULL)
> sample(10,3)
[1]  8 10  3
> set.seed(100)
> sample(10,3)
[1] 4 3 5
> set.seed(NULL)
> sample(10,3)
[1]  1 10  8
```

A veces querremos tomar diversas muestras aleatorias de una misma población y calcular algo sobre ellas. Para hacerlo podemos usar la función `replicate`. La sintaxis básica es

`replicate(n,`*instrucción*`)`

donde n es el número de repeticiones de la *instrucción*. Por ejemplo, para tomar 10 muestras aleatorias simples de 15 longitudes de pétalos de flores iris, podemos hacer:

```
> muestras=replicate(10, sample(iris$Petal.Length,15,
   replace=TRUE))
> muestras
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]  6.7  1.5  5.0  3.8  1.1  5.5  5.6  4.5  1.2   3.8
 [2,]  6.1  1.4  5.7  1.5  1.9  3.5  3.7  1.5  5.4   4.5
 [3,]  5.1  6.1  4.1  5.5  4.3  1.7  1.4  1.3  5.5   4.9
 [4,]  1.5  5.0  1.2  1.4  1.4  1.0  1.6  4.9  5.7   1.5
 [5,]  1.4  3.9  4.9  4.6  5.6  4.5  5.4  4.2  1.2   3.6
 [6,]  1.9  3.0  3.9  5.6  4.8  1.3  5.4  4.8  1.5   4.9
 [7,]  4.4  4.2  1.3  5.0  5.5  4.5  6.1  3.0  6.4   1.4
 [8,]  6.9  6.7  4.3  6.3  1.0  5.8  6.7  4.8  1.6   1.3
 [9,]  1.6  1.6  1.6  4.9  5.2  1.9  4.2  4.5  4.2   1.5
[10,]  6.4  4.2  1.3  1.4  1.4  6.7  5.0  1.5  1.5   5.5
[11,]  4.8  5.2  4.9  1.5  1.0  5.0  5.6  5.8  5.0   5.6
[12,]  1.4  6.7  4.9  5.3  3.9  4.3  4.2  6.0  1.6   4.1
[13,]  5.5  5.7  6.1  4.6  1.4  1.4  5.8  4.8  1.5   5.7
[14,]  1.6  4.7  1.5  5.1  1.2  1.4  5.1  1.5  5.6   1.5
[15,]  4.5  4.5  6.6  1.5  1.5  4.7  4.0  4.7  3.5   4.3
```

Observad que R ha organizado los diferentes resultados obtenidos con el `replicate` como columnas de una matriz.

Si, por ejemplo, sólo nos hubiera interesado calcular las medias, redondeadas a 2 cifras decimales, de 10 muestras aleatorias simples de 15 longitudes de pétalos de flores iris, podríamos haber hecho

```
> medias=replicate(10,round(mean(sample(iris$Petal.Length,15,
    replace=TRUE)),2))
> medias
 [1] 3.71 4.23 2.87 4.13 3.64 4.42 4.37 2.72 3.87 3.28
```

¿Y si quisiéramos la media y la desviación típica muestral de 10 muestras de estas? No podemos usar sin más dos `replicate`, porque las muestras serían diferentes. Pero tenemos dos opciones posibles. Una sería fijar la misma semilla de aleatoriedad antes de cada `replicate`.

```
> set.seed(1000)
> medias=replicate(10,round(mean(sample(iris$Petal.Length,15,
    replace=TRUE)),2))
> set.seed(1000)
> desv_tip=replicate(10,round(sd(sample(iris$Petal.Length,15,
    replace=TRUE)),2))
> rbind(medias,desv_tip)
          [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
medias    3.63 3.18 4.05 4.59 4.13 2.51 3.61 3.31 3.63  3.96
desv_tip 1.89 1.66 1.84 1.47 1.82 1.52 1.45 2.00 1.75  1.65
```

Otra posibilidad es definir una función que calcule un vector con estos dos valores, y luego usarla dentro del `replicate`. Veamos cómo, fijando la misma semilla de antes, nos da el mismo resultado.

```
> info=function(x){round(c(mean(x),sd(x)),2)}
> set.seed(1000)
> info_lp=replicate(10,info(sample(iris$Petal.Length,15,
   replace=TRUE)))
> info_lp
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,] 3.63 3.18 4.05 4.59 4.13 2.51 3.61 3.31 3.63  3.96
[2,] 1.89 1.66 1.84 1.47 1.82 1.52 1.45 2.00 1.75  1.65
```

En este último caso, R ha organizado por defecto la información obtenida como columnas de una matriz: la primera fila son las medias y la segunda las desviaciones típicas.

Como vemos, la función `set.seed` permite «fijar» el resultado de un `replicate` que incluya la generación de números aleatorios:

```
> set.seed(1000)
> replicate(10,round(mean(sample(iris$Petal.Length,15,
```

```
   replace=TRUE)),2))
 [1] 3.63 3.18 4.05 4.59 4.13 2.51 3.61 3.31 3.63 3.96
> set.seed(1000)
> replicate(10,round(mean(sample(iris$Petal.Length,15,
   replace=TRUE)),2))
 [1] 3.63 3.18 4.05 4.59 4.13 2.51 3.61 3.31 3.63 3.96
```

Guía rápida

- `sample(x, n, replace=...)` genera una muestra aleatoria de tamaño `n` del vector `x`, con reposición si igualamos `replace` a `TRUE` y sin reposición si lo igualamos a `FALSE`.
- `set.seed` permite fijar la semilla de aleatoriedad.
- `replicate(n,`*expresión*`)` evalúa `n` veces la *expresión*, y organiza los resultados como las columnas de una matriz.

Ejercicio

Consideremos la tabla de datos `datacrab.txt`, que encontraréis en el espacio virtual de la asignatura y que contiene información sobre una muestra de cangrejos. Cargadla en un *data frame*.

(a) Definid una función de parámetros s, n, m que calcule la media y la desviación típica del vector formado por las medias de los pesos de los individuos de cada una de las n muestras aleatorias simples de m (índices de) filas de dicha tabla obtenidas usando como semilla de aleatoriedad el número s. Tenéis que usar `set.seed` y `replicate` para definir la función.

(b) Aplicadla a $n = 50$, $m = 30$ y tomando como s el número formado por las 5 primeras cifras de vuestro NIF o pasaporte.

(c) ¿Qué valores predice el Teorema Central del Límite que se deberían obtener? ¿Habéis obtenido resultados similares a los predichos por dicho teorema?

Modelo de test

(1) Escogemos al azar 50 estudiantes de grado diferentes para preguntarles cuántas horas semanales estudian. ¿Qué tipo de muestreo hemos llevado a cabo?

   (a) Muestreo aleatorio simple

   (b) Muestreo aleatorio estratificado

(c) Muestreo aleatorio sin reposición

(d) Muestreo aleatorio por conglomerados

(e) Ninguno de los anteriores

(2) Con una sola instrucción, calculad la media de una muestra aleatoria sin reposición de 15 elementos escogidos de un vector numérico llamado X.

(3) Con una sola instrucción, extraed un subdataframe del dataframe `iris` formado por una muestra aleatoria sin reposición de 40 filas, y llamadlo `muestra`. Y antes de contestar, comprobad que funciona.

(4) Con una sola instrucción, calculad un vector formado por las medias de 100 muestras aleatorias sin reposición de 20 elementos cada una escogidos de un vector numérico llamado X y llamadlo `medias`.

Respuestas

(1) `(c)`

(2) `mean(sample(X,15,replace=FALSE))`
(También sería correcto, pero más complicado, `sum(sample(X,15,replace=FALSE))/15` Y en ambos casos también sería correcto sin `replace=FALSE`, puesto que este es el valor por defecto del parámetro.)

(3) `muestra=iris[sample(dim(iris)[1],40,replace=FALSE),]`
(También sería correcto mirar antes el número de filas con `str` o `tail`, ver que son 150, y responder `muestra=iris[sample(150,40,replace=FALSE),]`. Hay otras respuestas correctas, no las damos para no liaros. Además, y como antes, también sería correcto sin `replace=FALSE`, puesto que este es el valor por defecto del parámetro.)

(4) `medias=replicate(100,mean(sample(X,20,replace=FALSE)))`
(¿Ya os hemos dicho que también sería correcto sin `replace=FALSE`, puesto que este es el valor por defecto del parámetro?)