

Tablas y resúmenes estadísticos

Sitio: [Plataforma de Formación On line del Instituto Andaluz de
Administración Pública](#)
Curso: (I22F-PT05) Entorno de Programación R
Libro: Tablas y resúmenes estadísticos

Imprimido por: ALFONSO LUIS MONTEJO RAEZ
Día: lunes, 11 de abril de 2022, 10:55

Tabla de contenidos

1. Introducción
2. Tablas de frecuencias
3. Resúmenes estadísticos
4. Datos faltantes

1. Introducción

La principal misión de la estadística es extraer información de un conjunto de datos usando una serie de técnicas que van, desde la realización de análisis descriptivos a la inferencia (por ejemplo, la realización de contrastes de hipótesis).

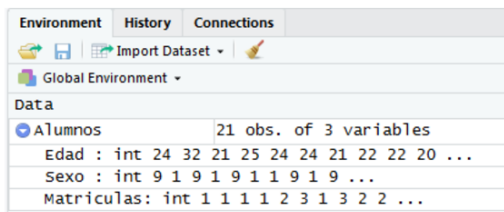
En este apartado vamos a ver como se realizan los análisis descriptivos básicos que podemos hacer a nuestros datos que, básicamente, son contar y ordenar los valores (**tablas**) y resumir su información en indicadores (**resúmenes estadísticos**).

Ejemplo

Para ilustrar el contenido de esta sección vamos a usar los datos del archivo **Alumnos.txt** que contiene información sobre la edad, sexo y el número de veces que se han matriculado en una asignatura un grupo de 21 alumnos y alumnas. Los datos dentro del fichero vienen separados por tabuladores.

Para ello lo primero es cargar el fichero en R. Podemos usar el asistente para la carga de datos en formato texto o usar la orden **read.delim()** que sirve para cargar datos con tabuladores.

```
> read.delim('F:\\Curso R IECA\\Temario\\Sesion2\\Datos\\Alumnos.txt')->Alumnos
```



Environment	History	Connections
Import Dataset		
Global Environment		
Data		
Alumnos 21 obs. of 3 variables		
Edad : int 24 32 21 25 24 24 21 22 22 20 ...		
Sexo : int 9 1 9 1 9 1 1 9 1 9 ...		
Matriculas: int 1 1 1 1 2 3 1 3 2 2 ...		

El comando `read.delim` sirve para leer ficheros de texto plano separado por tabuladores “\t”

Para tener las variables bien definidas tenemos que convertir la variable Sexo, que está definida como numérica, en una variable categórica (recordad que se llama Factor en R) donde el valor 1 sea Hombre y el 9 sea Mujer.

```
> factor(Alumnos$Sexo, levels=c(1,9), labels=c('Hombre', 'Mujer'))->Alumnos$Sexo2
```

Con esto ya tenemos cargados los datos en un objeto denominado Alumnos y la variable Sexo2 contiene la información categorizada del sexo del alumnado.

2. Tablas de frecuencias

Las **tablas de frecuencias** tratan de resumir información de un conjunto de datos simplemente contando su información y el número de casos de distinto tipo que aparece en cada variable.

Dentro de las tablas de frecuencias podemos encontrar dos tipos de recuentos: contar el número de apariciones de un valor (**frecuencia absoluta**) o expresar su número como porcentaje (**frecuencia relativa**).

Las órdenes en R para obtener ambas tablas serían **table()** para las frecuencias absolutas y **prop.table()** para las relativas. Esta última función es un poco especial ya que no podemos usarla directamente sobre una variable, si no que siempre se tiene que usar sobre una tabla, es decir, **prop.table(table(variable))**.

Con estas funciones también se pueden obtener tablas de frecuencias absolutas de dos o más variables (distribuciones conjuntas), simplemente separando con comas las que queremos que estén en la tabla. Hay que tener en cuenta que la primera variable elegida siempre irá en las filas de la tabla.

En el caso de las tablas de frecuencias relativas se pueden obtener tres análisis distintos:

- **Respecto al total.** Calcula la proporción respecto a todos los valores de la tabla. Se obtiene usando **prop.table(vble1,vble2)**.
- **Por columna.** Calcula la proporción respecto al total de la variable situada en la columna. Se obtiene usando **prop.table(vble1,vble2,2)**.
- **Por fila.** Calcula la proporción respecto al total de la variable situada en la fila. Se obtiene usando **prop.table(vble1,vble2,1)**.

Ejemplo

Usando los datos cargados en la introducción, vamos a obtener la tabla de frecuencia absoluta de la variable edad y la frecuencia relativa de la variable Sexo2 con las órdenes **table()** y **prop.table()**.

```
> table(Alumnos$Edad)
```

```
18 19 20 21 22 23 24 25 32
 1  3  3  3  2  2  3  2  1
```

```
> prop.table(table(Alumnos$Sexo2))
```

```
      Hombre      Mujer
0.5238095 0.4761905
```

En este caso podemos ver el número de personas clasificadas según su edad y la proporción de personas según el sexo (si queremos expresarlos en porcentaje tenemos que multiplicar por 100).

También podemos crear una tabla de doble entrada o tabla de contingencia, en este caso, de las variables Sexo2 y N° de veces matriculado usando **table()** y separando las variables con comas. Recordad que la primera variable será la que aparezca en las filas.

```
> table(Alumnos2$Sexo2,Alumnos$Matriculas)
```

```
      1 2 3
Hombre 6 4 1
Mujer  4 5 1
```

También podemos almacenar una tabla como un objeto y calcular la frecuencia relativa conjunta (de dos variables) con la orden **prop.table()**.

```
> table(Alumnos2$Sexo2,Alumnos$Matriculas)->tabla
> prop.table(tabla)
```

```
      1      2      3
Hombre 0.28571429 0.19047619 0.04761905
Mujer  0.19047619 0.23809524 0.04761905
```

Aquí tenemos la información de manera relativa respecto al total. Si queremos la información en porcentaje hay que multiplicar por 100 el contenido de la tabla.

```
> prop.table(tabla,1)
```

	1	2	3
Hombre	0.54545455	0.36363636	0.09090909
Mujer	0.40000000	0.50000000	0.10000000

```
> prop.table(tabla,2)
```

	1	2	3
Hombre	0.6000000	0.4444444	0.5000000
Mujer	0.4000000	0.5555556	0.5000000

En la primera tabla tenemos la proporción por filas (el 40% de las mujeres se ha matriculado una sola vez en la asignatura) y en la segunda la proporción por columnas (el 50% de las personas matriculadas tres veces son hombres).

3. Resúmenes estadísticos

Los indicadores o resúmenes estadísticos son medidas que nos permiten obtener información sobre varios aspectos de un conjunto de datos.

En general la información que podemos obtener se puede agrupar en cuatro tipos:

- **Centralización.** Indican valores con respecto a los que los datos parecen agruparse. Media, mediana y moda.
- **Dispersión.** Indican la mayor o menor concentración de los datos con respecto a las medidas de centralización. Desviación típica, coeficiente de variación, rango, varianza.
- **Posición.** Dividen un conjunto ordenado de datos en grupos con la misma cantidad de individuos. Cuantiles (percentiles, cuartiles, deciles).
- **Forma.** Asimetría, Apuntamiento o curtosis.

Las funciones estadísticas básicas en R serían:

- **mean().** Media
- **median().** Mediana
- **var().** Cuasivarianza
- **sd().** Cuasidesviación típica
- **quantile()** o **fivenum().** Calcula los cuartiles
- **summary().** Resumen estadístico

Si queremos conocer todas las funciones estadísticas que tiene instaladas R en el paquete base podemos usar la siguiente función

library(help='stats').

También existen otros paquetes disponibles que ofrecen resúmenes estadísticos como **agricolae** o **psych** pero tenemos que instalarlos en el caso de que no los tengamos.

Ejemplo

Podemos usar la función **summary()** para obtener un resumen con los principales indicadores estadísticos descriptivos.

```
> summary(Alumnos$Edad)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 18.0   20.0   21.5   22.1   24.0   32.0      1

> summary(Alumnos$Sexo2)
Hombre  Mujer
    11     10
```

La particularidad de la orden **summary()** es que se adapta a cada tipo de objeto y nos va ofreciendo una información distinta en función de cada tipo. Si es numérico (como la variable Edad) nos dice sus principales estadísticos descriptivos (máximo, cuartiles, mínimo y media) y si es un factor (como la variable Sexo2) nos hace una tabla de frecuencias absolutas.

También podemos usar otros paquetes como por ejemplo **psych**. Si hacemos un resumen estadístico del objeto alumnos usando la orden **describe()** del paquete **psych** obtenemos una tabla con los principales indicadores de todas las variables que tiene el objeto.

```
> psych::describe(Alumnos2)
      vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
Edad      1 21 22.10 3.08    22   21.77 2.97   18 32    14 1.39    2.50 0.67
Sexo      2 21  4.81 4.09     1    4.76 0.00    1  9     8 0.09   -2.08 0.89
Matriculas 3 21  1.62 0.67     2    1.53 1.48    1  3     2 0.54   -0.88 0.15
Sexo2*    4 21  1.48 0.51     1    1.47 0.00    1  2     1 0.09   -2.08 0.11
```

En este ejemplo hemos usado una forma distinta de llamar una función de un paquete.

Podemos usar una orden, que se encuentre en un paquete instalado por nosotros, sin necesidad de tenerlo cargado, usando los dos puntos dobles de esta manera:

paquete::orden()

Esto es útil cuando solo vamos a usar una función concreta y así nos ahorramos el espacio en memoria que supone cargar el paquete entero. Lo que si es obligatorio es que el paquete debe estar instalado en nuestro ordenador ya que si no, la orden no funcionaría.

En el ejemplo hemos usado la orden **psych::describe(Alumnos2)** para usar la función describe() del paquete psych.

4. Datos faltantes

Uno de los principales problemas, cuando estamos haciendo tablas o resúmenes de datos, son los **datos faltantes o missing values**. Este sería un dato que, por el motivo que fuera, no se ha podido conseguir con lo que no está disponible para trabajar con él. En el caso de R los marca como NA.

Cuando aparecen este tipo de datos hay algunas funciones que no se pueden calcular y devuelven un valor NA.

Para evitar los problemas de los valores perdidos hay varias formas de actuar:

- **Ignorar esos valores.** Todas las órdenes de R tienen un parámetro para ignorar los valores NA pero en cada orden se usa de manera distinta.
- **Imputar el valor.** Se trataría de asignar un nuevo valor para sustituir el valor que falta. Normalmente se usa la media de los valores de la variable, aunque se pueden implementar otras soluciones.
- **Eliminar el registro.** Se trataría de eliminar toda la información aportada por la persona a la que le falta el dato, eliminando la fila completa.

Para detectar si existen valores faltantes en una variable se usa la función **is.na()**, que nos devuelve un vector lógico con los valores TRUE si el valor está perdido y FALSE si el valor existe, o **summary()** que nos indica el número de NA que tiene una variable.

Ejemplo

Con los datos de alumnado con los que estamos trabajando vamos a buscar los valores perdidos en la variable Edad.

```
> mean(Alumnos$Edad)
[1] NA
```

Si calculamos la media de la variable edad nos devuelve un NA. Esto es un claro indicio de que existen valores perdidos en la variable.

Para detectarlos podemos usar la función **is.na()** o **summary()**.

```
> is.na(Alumnos$Edad)
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[14] FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE

> summary(Alumnos$Edad)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
  18.0   20.0   21.5   22.1   24.0   32.0     1
```

En el primer caso observamos que hay un valor perdido (TRUE) en la posición 17 y en el segundo caso vemos que existe un valor perdido (NA's 1).

A partir de este momento podemos seguir varias estrategias para trabajar con este tipo de valores.

Ignorar el valor. Todas las funciones tienen un parámetro para ignorar los valores perdidos.

```
> table(Alumnos$Edad, useNA='ifany')

 18   19   20   21   22   23   24   25   32 <NA>
  1    3    3    3    2    2    3    2    1     1

> mean(Alumnos$Edad, na.rm=T)
[1] 22.1
```

En el caso de la función **table()** se usa el parámetro **useNA** y en el caso de **mean()** sería el parámetro **na.rm**. Para saber en cada orden cuál es el parámetro que debemos usar tenemos que consultar la ayuda de dicha orden en la pestaña **Help**.

Imputar el valor. Sustituir el valor por el valor medio

```
> Alumnos->Alumnos2
> Alumnos2[17,1]<-mean(Alumnos2$Edad,na.rm=T)
> summary(Alumnos2$Edad)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 18.0   20.0   22.0   22.1   24.0   32.0
```

En este caso duplicamos el objeto Alumnos como Alumnos2, ésta es una buena opción si queremos mantener el objeto original (Alumnos) sin modificar por si algo sale mal.

Le asignamos al objeto Alumnos2 en la fila 17 (donde está el dato NA), columna 1 (donde está la variable Edad) el valor de la media de edad del resto de los datos de la variable. Al hacer de nuevo el summary() ya no aparece el dato faltante.

Eliminar el registro. Localizamos el valor faltante y eliminamos la fila completa.

```
> Alumnos[is.na(Alumnos$Edad)==T,]
  Edad Sexo Matriculas  Sexo2
17  NA    1           1 Hombre

> Alumnos[-17,]->Alumnos_nona
> summary(Alumnos_nona$Edad)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 18.0   20.0   21.5   22.1   24.0   32.0
```

Primero filtramos el objeto Alumnos para saber que fila es la que contiene el valor perdido (usando la función **is.na()**) y generamos un nuevo objeto (Alumnos_nona) sin esa fila. Cuando hacemos el **summary()** ya no está el dato faltante.

Esta estrategia hay que tomarla con cautela ya que estamos perdiendo información de esa misma fila que sí estaba rellena para otras variables.