Manual de R

Freddy Hernández Barajas
Olga Cecilia Usuga Manco

2016-12-03

Índice general

Prefacio

En este manual de R se explican de forma sencilla las funciones y procedimientos básicos para un análisis estadístico. Como complemento a este libro se recomienda consulta Correa (2016) para la construcción de gráficos en R.

Capítulo 1

Medidas de tendencia central

En este capítulo se mostrará cómo obtener las diferentes medidas de tendencia central con R.

Para ilustrar el uso de las funciones se utilizará una base de datos llamada **medidas del cuerpo**, esta base de datos cuenta con 6 variables registradas a un grupo de 36 estudiantes de la universidad. Las variables son:

1. `edad` del estudiante (años),
2. `peso` del estudiante (kilogramos),
3. `altura` del estudiante (centímetros),
4. `sexo` del estudiante (Hombre, Mujer),
5. `muneca`: perímetro de la muñeca derecha (centímetros),
6. `biceps`: perímetro del biceps derecho (centímetros).

A continuación se presenta el código para definir la url donde están los datos, para cargar la base de datos en R y para mostrar por pantalla un encabezado (usando `head`) de la base de datos.

```
url <- 'https://raw.githubusercontent.com/fhernanb/datos/master/medidas_cuerpo'
datos <- read.table(file=url, header=T)
head(datos)  # Para ver el encabezado de la base de datos
```

```
##   edad peso altura   sexo muneca biceps
## 1   43 87.3  188.0 Hombre   12.2   35.8
## 2   65 80.0  174.0 Hombre   12.0   35.0
## 3   45 82.3  176.5 Hombre   11.2   38.5
## 4   37 73.6  180.3 Hombre   11.2   32.2
## 5   55 74.1  167.6 Hombre   11.8   32.9
## 6   33 85.9  188.0 Hombre   12.4   38.5
```

1.1. Media

Para calcular la media de una variable cuantitativa se usa la función `mean`. Los argumentos básicos de la función `mean` son dos y se muestran a continuación.

```
mean(x, na.rm)
```

Ejemplo

Suponga que queremos obtener la altura media del grupo de estudiantes.

Para encontrar la media general se usa la función `mean` sobre el vector númerico `datos$altura`.

```
mean(x=datos$altura)
```

```
## [1] 171.5556
```

Del anterior resultado podemos decir que la estatura media o promedio de los estudiantes es 171.5555556 centímetros.

Ejemplo

Suponga que ahora queremos la altura media pero diferenciando por sexo.

Para hacer esto se debe primero dividir o partir el vector de altura según los niveles de la variable sexo, esto se consigue por medio de la función `split` y el resultado será una lista con tantos elementos como niveles tenga la variable sexo. Luego a cada uno de los elementos de la lista se le aplica la función `mean` con la ayuda de `sapply` o `tapply`. A continuación el código completo para obtener las alturas medias para hombres y mujeres.

```
sapply(split(x=datos$altura, f=datos$sexo), mean)
```

```
##   Hombre    Mujer
## 179.0778 164.0333
```

El resultado es un vector con dos elementos, vemos que la altura media para hombres es 179.0777778 centímetros y que para las mujeres es de 164.0333333 centímetros.

¿Qué sucede si se usa `tapply` en lugar de `sapply`? Substituya en el código anterior la función `sapply` por `tapply` y observe la diferencia entre los resultados.

Ejemplo

Suponga que se tiene el vector `edad` con las edades de siete personas y supóngase que para el individuo cinco no se tiene información de su edad, eso significa que el vector tendrá un `NA` en la quinta posición.

¿Cuál será la edad promedio del grupo de personas?

```
edad <- c(18, 23, 26, 32, NA, 32, 29)
mean(x=edad)
```

```
## [1] NA
```

Al correr el código anterior se obtiene un error y es debido al símbolo `NA` en la quinta posición. Para calcular la media sólo con los datos de los cuales se tiene información, se incluye el argumento `na.rm = TRUE` para que R remueva los `NA`. El código correcto a usar en este caso es:

```
mean(x=edad, na.rm=TRUE)
```

```
## [1] 26.66667
```

De este último resultado se obtiene que la edad promedio de los individuos es 26.67 años.

1.2. Mediana

Para calcular la mediana de una variable cantitativa se usa la función `median`. Los argumentos básicos de la función `median` son dos y se muestran a continuación.

```
median(x, na.rm)
```

Ejemplo

Calcular la edad mediana para los estudiantes de la base de datos.

Para obtener la mediana usamos el siguiente código:

```
median(x=datos$edad)
```

```
## [1] 28
```

y obtenemos que la mitad de los estudiantes tienen edades mayores o iguales a 28 años.

El resultado anterior se pudo haber obtenido con la función `quantile` e indicando que se desea el cuantil 50 así:

```
quantile(x=datos$edad, probs=0.5)
```

```
## 50%
##  28
```

1.3. Moda

La moda de una variable cuantitativa corresponde a valor o valores que más se repiten, una forma sencilla de encontrar la moda es construir una tabla de frecuencias y observar los valores con mayor frecuencia.

Ejemplo

Calcular la moda para la variable edad de la base de datos de estudiantes.

Se construye la tabla con la función `table` y se crea el objeto `tabla` para almacenarla.

```
tabla <- table(datos$edad)
tabla
```

```
##
## 19 20 21 22 23 24 25 26 28 29 30 32 33 35 37 40 43 45 51 55 65
##  1  1  1  3  2  1  5  3  2  1  2  1  1  2  3  1  2  1  1  1  1
```

Al mirar con detalle la tabla anterior se observa que el valor que más se repite es la edad de 25 años en 5 ocasiones. Si la tabla hubiese sido mayor, la inspección visual nos podría tomar unos segundos o hasta minutos y podríamos equivocarnos, por esa razón es mejor ordenar los resultados de la tabla.

Para observar los valores con mayor frecuencia de la tabla se puede ordenar la tabla usando la función `sort` de la siguiente manera:

```
sort(tabla, decreasing=TRUE)
```

```
##
## 25 22 26 37 23 28 30 35 43 19 20 21 24 29 32 33 40 45 51 55 65
##  5  3  3  3  2  2  2  2  2  1  1  1  1  1  1  1  1  1  1  1  1
```

De esta manera se ve fácilmente que la variable edad es unimodal con valor de 25 años.

Capítulo 2

Medidas de variabilidad

En este capítulo se mostrará cómo obtener las diferentes medidas de variabilidad con R.

Para ilustrar el uso de las funciones se utilizará la base de datos llamada **aptos2015**, esta base de datos cuenta con 11 variables registradas a apartamentos usados en la ciudad de Medellín. Las variables son:

1. `precio`: precio de venta del apartamento (millones de pesos),
2. `mt2`: área del apartamento (m^2),
3. `ubicación`: lugar de ubicación del aparamentos en la ciudad (cualitativa),
4. `estrato`: nivel socioeconómico donde está el apartamento (2 a 6),
5. `alcobas`: número de alcobas del apartamento,
6. `banos`: número de baños del apartamento,
7. `balcon`: si el apartamento tiene balcón (si o no),
8. `parqueadero`: si el apartamento tiene parqueadero (si o no),
9. `administracion`: valor mensual del servicio de administración (millones de pesos),
10. `avaluo`: valor del apartamento en escrituras (millones de pesos),
11. `terminado`: si el apartamento se encuentra terminado (si o no).

A continuación se presenta el código para definir la url donde están los datos, para cargar la base de datos en R y para mostrar por pantalla un encabezado (usando `head`) de la base de datos.

```
url <- 'https://raw.githubusercontent.com/fhernanb/datos/master/aptos2015'
datos <- read.table(file=url, header=T)
head(datos)  # Para ver el encabezado de la base de datos
```

```
##   precio   mt2 ubicacion estrato alcobas banos balcon parqueadero
## 1     79 43.16     norte       3       3     1     si          si
## 2     93 56.92     norte       2       2     1     si          si
## 3    100 66.40     norte       3       2     2     no          no
## 4    123 61.85     norte       2       3     2     si          si
## 5    135 89.80     norte       4       3     2     si          no
## 6    140 71.00     norte       3       3     2     no          si
##   administracion   avaluo terminado
## 1          0.050 14.92300        no
## 2          0.069 27.00000        si
## 3          0.000 15.73843        no
## 4          0.130 27.00000        no
## 5          0.000 39.56700        si
## 6          0.120 31.14551        si
```

2.1. Rango

Para calcular el rango de una variable cuantitativa se usa la función `range`. Los argumentos básicos de la función `range` son dos y se muestran abajo.

```
range(x, na.rm)
```

La función `range` entrega el valor mínimo y máximo de la variable ingresada y el valor de rango ($max - min$) se puede obtener restando del máximo el mínimo.

Ejemplo

Suponga que queremos obtener el rango para la variable precio de los apartamentos.

Para obtener el rango usamos el siguiente código.

```
range(datos$precio)
```

```
## [1]   25 1700
```

```
max(datos$precio) - min(datos$precio)
```

```
## [1] 1675
```

Del resultado anterior podemos ver que los precios de todos los apartamentos van desde 25 hasta 1700 millones de pesos, es decir, el rango de la variables precio es 1675 millones de pesos.

Ejemplo

Suponga que queremos obtener nuevamente el rango para la variable precio de los apartamentos pero diferenciando por el estrato.

Primero vamos a crear una función auxiliar llamada `myrange` que calculará el rango directamente ($max - min$). Luego vamos a partir la información de los precios por cada estrato usando `split`, la partición se almacenará en la lista `precios`. Finalmente se aplicará la función `myrange` a la lista `precios` para obtener los rangos del precio por estrato socioeconómico. El código para realizar esto se muestra a continuación.

```
myrange <- function(x) max(x) - min(x)
precios <- split(datos$precio, f=datos$estrato)
sapply(precios, myrange)
```

```
##    2    3    4    5    6
##  103  225  610 1325 1560
```

De los resultados podemos ver claramente que a medida que aumenta de estrato el rango (variabilidad) del precio de los apartamentos aumenta. Apartamentos de estrato bajo tienden a tener precios similares mientras que los precios de venta para apartamentos de estratos altos tienden a ser muy diferentes entre si.

Capítulo 3

Medidas de posición

En este capítulo se mostrará cómo obtener las diferentes medidas de

Capítulo 4

Medidas de correlación

En este capítulo se mostrará cómo obtener las diferentes medidas de

Capítulo 5

Funciones básicas de R

En este capítulo se

Capítulo 6

Creación de funciones en R

En este capítulo se

Capítulo 7

Distribuciones discretas

En este capítulo se

Capítulo 8

Distribuciones continuas

En este capítulo se

Capítulo 9

Distribuciones continuas

En este capítulo se

Capítulo 10

Aproximación de integrales

En este capítulo se mostrará cómo aproximar integrales en una y varias dimensiones.

10.1. Aproximación de Laplace unidimensional

Esta aproximación es útil para obtener el valor de una integral usando la expansión de Taylor para una función $f(x)$ unimodal en $\Re$, en otras palabras lo que interesa es:

$$I = \int_{-\infty}^{\infty} f(x)d(x)$$

Al hacer una expansión de Taylor de segundo orden para $\log(f(x))$ en su moda x_0 el resultado es:

$$\log(f(x)) \approx \log(f(x_0)) + \frac{\log(f)'(x_0)}{1!}(x - x_0) + \frac{\log(f)''(x_0)}{2!}(x - x_0)^2$$

El segundo término de la suma se anula porque $\log(f)'(x_0) = 0$ por ser x_0 el valor donde está el máximo de $\log(f(x))$. La expresión anterior se simplifica en:

$$\log(f(x)) \approx \log(f(x_0)) + \frac{\log(f)''(x_0)}{2!}(x - x_0)^2$$

al aislar $f(x)$ se tiene que

$$f(x) \approx f(x_0) \exp\left(-\frac{c}{2}(x - x_0)^2\right) \tag{10.1}$$

donde $c = -\frac{d^2}{dx^2} \log(f(x))\Big|_{x=x_0}$.

La expresión 10.1 se puede reescribir de manera que aparezca el núcleo de la función de densidad de la distribución normal con media x_0 y varianza $1/c$, a continuación la expresión

$$f(x) \approx f(x_0) \frac{\sqrt{2\pi/c}}{\sqrt{2\pi/c}} \exp\left(-\frac{1}{2}\left(\frac{x - x_0}{1/\sqrt{c}}\right)^2\right)$$

Así al calcular la integral de $f(x)$ en $\Re$ se tiene que:

$$I = \int_{-\infty}^{\infty} f(x)d(x) = f(x_0)\sqrt{2\pi/c} \tag{10.2}$$

10.1.1. Ejemplo

Calcular la integral de $f(x) = \exp\left(-(x-1.5)^2\right)$ en $\Re$ utilizando la aproximación de Laplace.

Primero vamos a dibujar la función $f(x)$ para ver en dónde está su moda x_0.

```
fun <- function(x) exp(-(x-1.5)^2)
curve(fun, from=-5, to=5, ylab='f(x)', las=1)
```

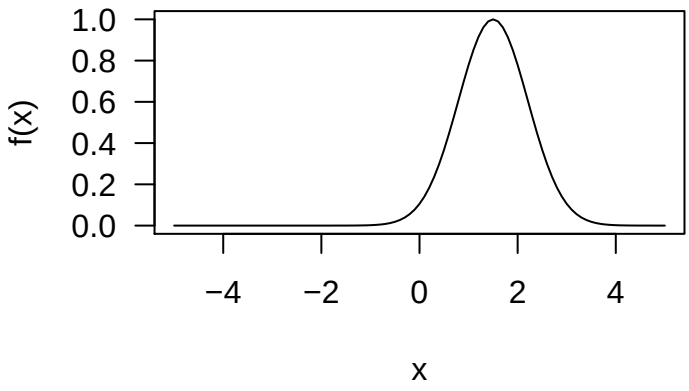

Figura 10.1: Perfil de la función f(x).

Visualmente se nota que la moda está cerca del valor 1.5 y para determinar numéricamente el valor de la moda x_0 se usa la función `optimize`, los resultados se almacenan en el objeto `res`. El valor de la moda corresponde al elemento `maximum` del objeto `res`.

```
res <- optimize(fun, interval=c(-10, 10), maximum=TRUE)
res
```

```
## $maximum
## [1] 1.499997
##
## $objective
## [1] 1
```

Para determinar el valor de c de la expresión 10.2 se utiliza el siguiente código.

```
require("numDeriv")
constant <- - as.numeric(hessian(fun, res$maximum))
```

Para obtener la aproximación de la integral se usa la expresión 10.2 y para tener un punto de comparación se evalua la integral usando la función `integrate`, a continuación el código.

```
fun(res$maximum) * sqrt(2*pi/constant)
```

```
## [1] 1.772454
```

```
integrate(fun, -Inf, Inf)  # Para comparar
```

```
## 1.772454 with absolute error < 1.5e-06
```

De los anteriores resultados vemos que la aproximación es buena.

Bibliografía

Correa, J. C. & Hernández, F. (2016). *Gráficos en R*. UNAL, Medellín, Colombia, 1ed edition. ISBN xxxxxx.

Índice alfabético