

Módulo 1. Introducción a R

Curso Herramientas para Data Science

Prof. Jose Jacobo Zubcoff Vallejo, PhD
Universidad de Alicante

2017 @ Licencia Creative Common BY

Módulo 1. Introducción a R

Lección 1. El entorno de R. Comandos, resultados y sesiones

R el lenguaje y su aplicación

La aplicación para distintos sistemas operativos se puede descargar de la página oficial “Comprehensive R Archive Network” (CRAN) <https://cran.r-project.org>. En dicha página se encuentran también tutoriales y documentos de ayuda.

Una vez descargada e instalada la aplicación tiene una interfaz similar en todos los sistemas operativos (Windows, Mac OS X, Unix y Linux).

El entorno de R

El entorno consta de una ventana llamada “Consola” que es donde se deben introducir los comandos del lenguaje R y donde aparecerán los resultados no gráficos.

```
> x <- 1 #asignamos el valor 1 a 'x'
> x #ver el contenido de 'x'
[1] 1
>
> x <- 2*3 #asignamos 6 a 'x' (resultado de multiplicar 2 * 3)
> x #ver el contenido de 'x'
[1] 6
> mode(x) #Devuelve el tipo de objeto de 'x'
[1] "numeric"
>
> y <- TRUE #asignamos el valor TRUE o verdadero a 'y'
> mode(y) #Devuelve el tipo de objeto de 'y'
[1] "logical"
> y
[1] TRUE
> #En la consola de R los comandos tienen un color y los resultados de R tienen otro color
> z <- c(1,2,3) #Asigna tres datos a un vector 'z'
> z
[1] 1 2 3
> |
```

Figure 1: Consola de R (en Mac OS X)

Los gráficos aparecen en otra ventana (denominada Quartz en Mac OS X). Cada vez que se hace un gráfico, se reemplaza el gráfico anterior, no se guarda automáticamente el historial de gráficos. Debemos hacerlo manualmente seleccionando la ventana y eligiendo la opción de menú correspondiente.

Comandos, sus resultados y comentarios

En la consola debemos introducir los comandos para obtener un resultado de R. El símbolo `>` en la consola nos indica que está listo para trabajar, sin ese símbolo estará haciendo alguna tarea y habrá que esperar que termine tal tarea (o forzar su finalización). Podemos usar R como calculadora, es en realidad una potente calculadora. Por ejemplo, si queremos multiplicar $3*7$

```
3*7
```

```
## [1] 21
```

Nos indica que el resultado de multiplicar $3*7$ es 21. En este ejemplo podemos observar que los comandos que introducimos en R tienen un color, mientras que los resultados de R tienen otro color (dependerá de la configuración y del Sistema Operativo los colores de cada uno). Otra característica de R es que el resultado lo ha devuelto con un `[1]` delante. Eso es porque R prefiere entender que es un vector de resultados, pero que solo tiene un resultado en esta ocasión.

Por tanto la consola es nuestra interfaz con R. Cada vez que introduzcamos un comando, R devolverá una respuesta. En el caso en que no hayamos terminado de introducir el comando, por ejemplo en la multiplicación anterior no hemos puesto en la misma línea el 7

```
3*  
7
```

```
## [1] 21
```

R esperará que en la siguiente línea le introduzcamos por cuánto queremos multiplicar el 3. La forma que tiene de avisarnos es con el símbolo `+`

Los primeros comandos que usaremos son los que sirven para crear una variable y asignarle un valor, ya sea numérico, texto o lógico. Se puede asignar a izquierda (punta de la flecha apuntando hacia la izquierda) o a derecha. Por ejemplo, algunas asignaciones a izquierda.

```
x <- 1          # asignamos el valor 1 a 'x'  
x              # ver el contenido de 'x'
```

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

```
x <- 2*3        # asignamos 6 a 'x' (resultado de multiplicar 2 * 3)  
x              # ver el contenido de 'x'
```

```
## [1] 6
```

```
mode(x)        # devuelve el tipo de objeto de 'x'
```

```
## [1] "numeric"
```

```
y <- TRUE       # asignamos el valor TRUE o verdadero a 'y'  
mode(y)        # devuelve el tipo de objeto de 'y'
```

```
## [1] "logical"
```

```
y
```

```
## [1] TRUE
```

También podemos asignar a derecha:

```
FALSE -> y          # asignamos el valor FALSE o falso a 'y'
mode(y)             # devuelve el tipo de objeto de 'y'
```

```
## [1] "logical"
y
```

```
## [1] FALSE
```

```
"a" -> y            # asignación a derecha de y="a"
mode(y)             # devuelve el tipo de objeto de 'y'
```

```
## [1] "character"
y
```

```
## [1] "a"
```

En R es frecuente trabajar con vectores, que son objetos que contienen varios valores (numéricos, caracteres o lógicos).

```
z <- c(1,2,3)        # asigna tres datos a un vector 'z'
mode(z)
```

```
## [1] "numeric"
z
```

```
## [1] 1 2 3
```

```
misnums <- seq(from=1,to=5,by=1) # asigna los numeros del 1 al 5 al objeto 'minum'
misnums                                     # ver el contenido de 'misnums'
```

```
## [1] 1 2 3 4 5
```

```
mode(misnums)          # devuelve el tipo de objeto de 'misnums'
```

```
## [1] "numeric"
```

Tenemos un vector con los números del 1 al 5 almacenados en ‘misnums’.

Sesiones, histórico de comandos y datos en memoria

En R, las variables se almacenan en memoria mientras dure la sesión (ventana abierta). Para guardar nuestra sesión para poder reproducirla en otro momento o en otro ordenador, podemos guardar todo el histórico de comandos usados en la sesión en un fichero, por ejemplo crearemos el fichero “MisComandos.Rhistory”. Para ello, usaremos el comando:

```
savehistory(file = "MisComandos.Rhistory")
#savehistory(file = "MisComandos.Rhistory")
```

Gráficas simples

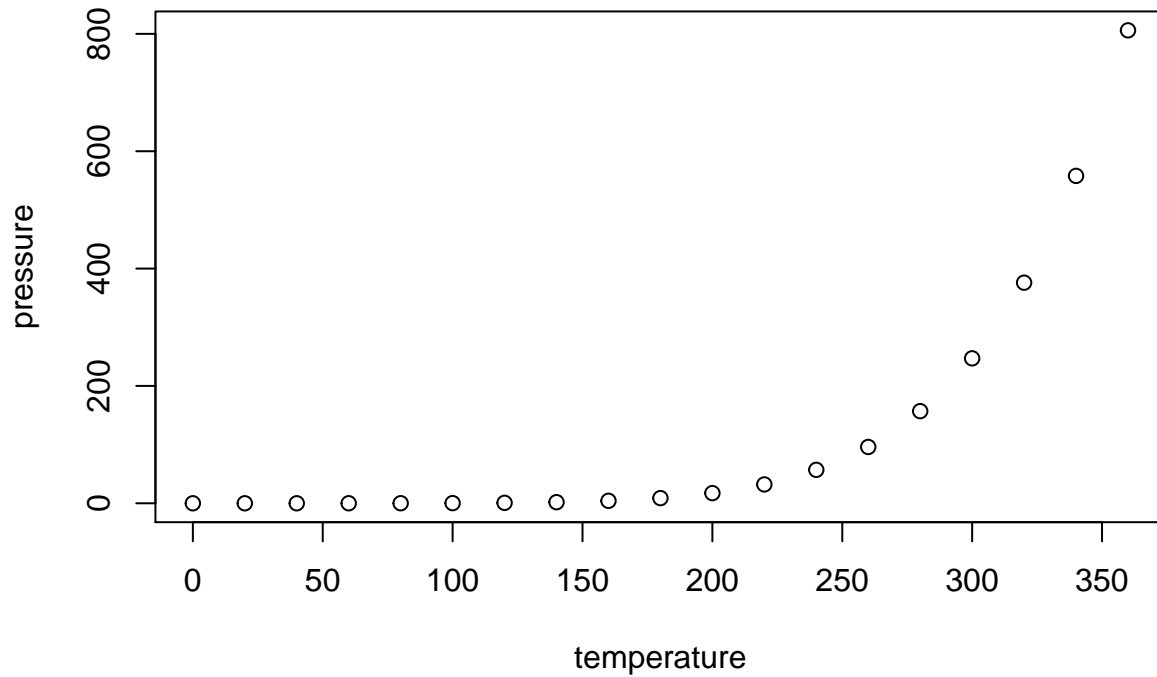
Diagramas de puntos

Podemos hacer un gráfico de unos datos que se hayan cargado en memoria (importando datos o introduciendo manualmente). Vamos a usar los datos que ya incorpora R sobre presión (“pressure”) a modo de ejemplo.

```
head(pressure)
```

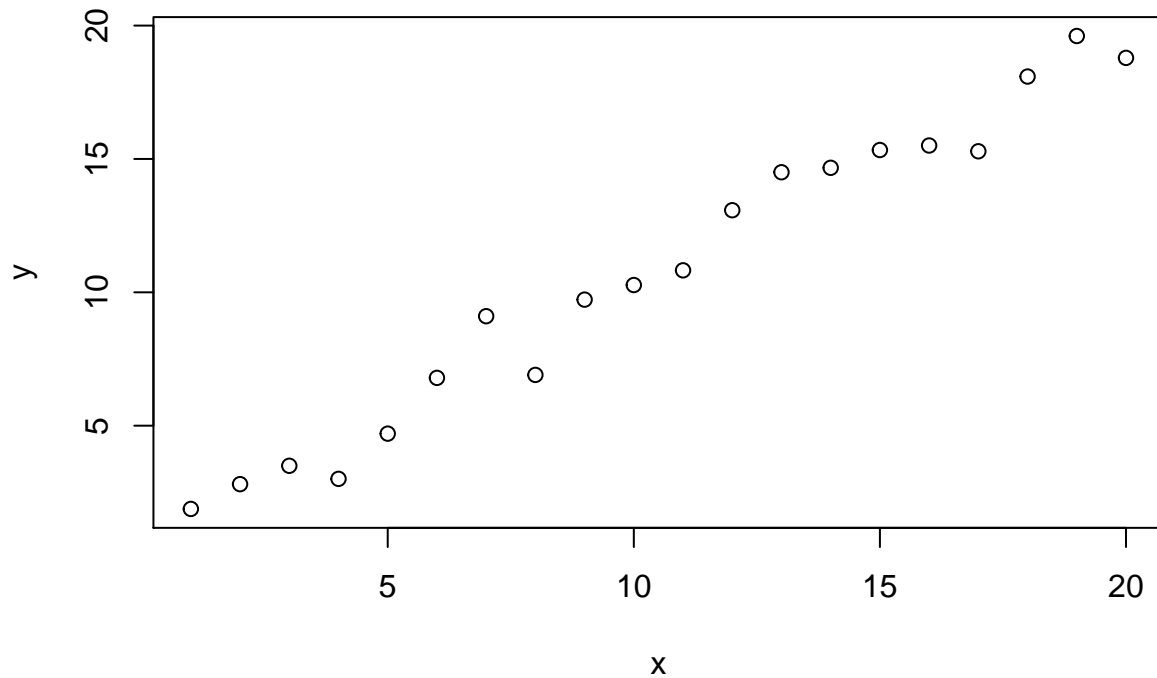
```
## temperature pressure
## 1      0  0.0002
## 2     20  0.0012
## 3     40  0.0060
## 4     60  0.0300
## 5     80  0.0900
## 6    100  0.2700
```

```
plot(pressure)
```



Ahora crearemos unos datos aleatorios para poder hacer algunas gráficas con ellos.

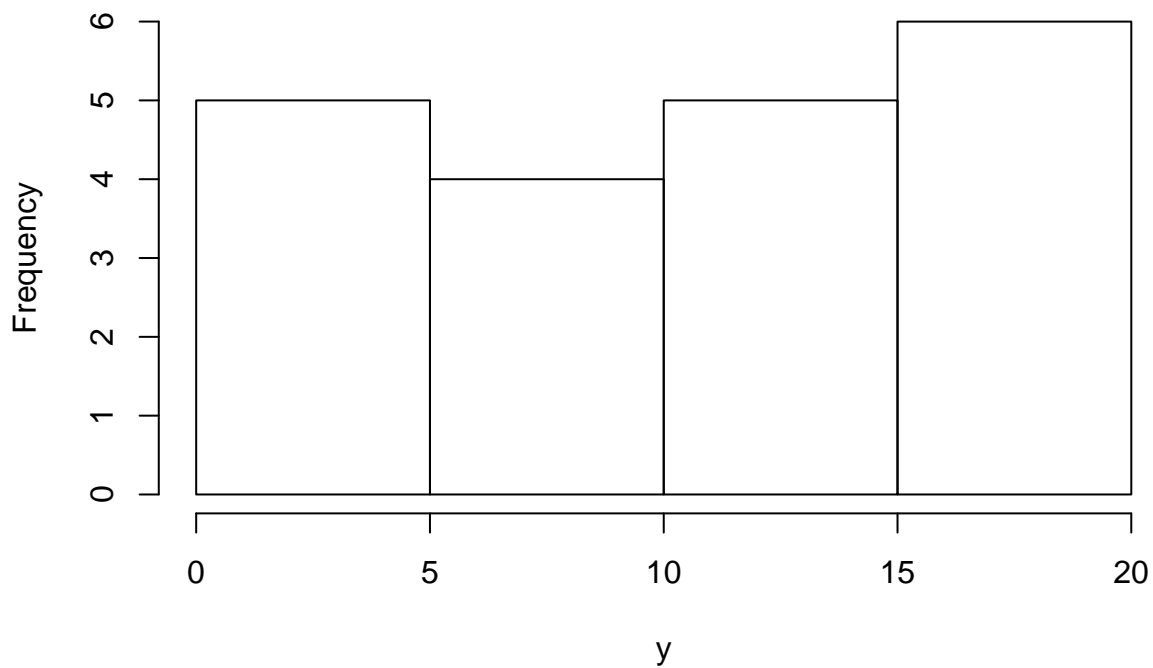
```
x <- 1:20          # asignamos a x los valores de 1 a 20
y <- x + rnorm(x)  # rnorm() genera datos aleatorios con probabilidad normal
plot(x,y)          # nube de puntos de x,y
```



Podemos hacer un histograma muy fácilmente con R. Solo necesitamos pasar la variable que contiene los datos a la función “hist()”.

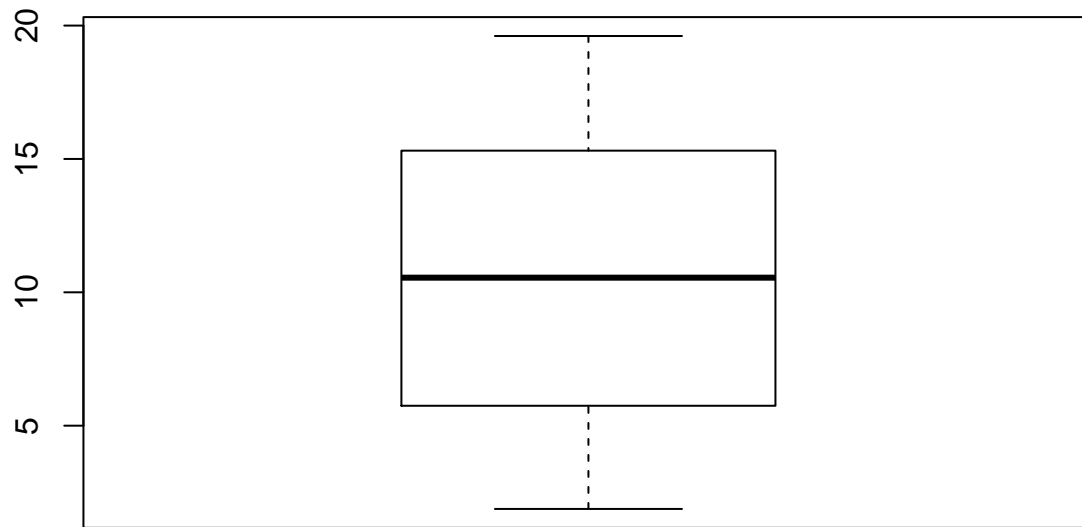
```
hist(y) #histograma de y
```

Histogram of y



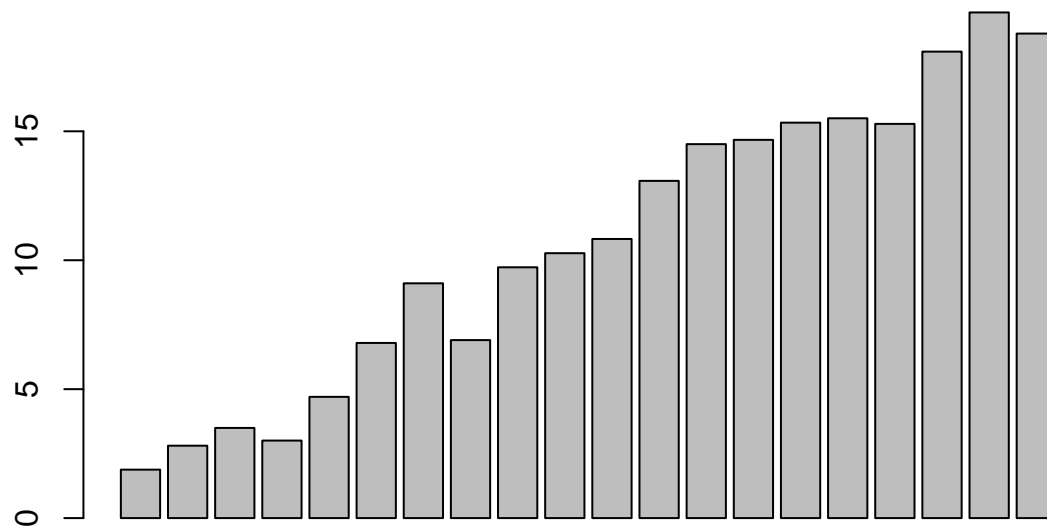
También podemos dibujar un diagrama de cajas o boxplot con la función “boxplot()”.

```
boxplot(y) #Diagrama de cajas de y
```



La función para dibujar diagramas de barra en R es “`barplot()`”.

```
barplot(y) #Diagrama de barras de y
```



También podemos dibujar diagramas tipo tarta, o quesitos en R con la función “`pie()`”.

```
pie(x) #Diagrama de tartas de x
```

