

The R logo is a large, stylized graphic in the background of the slide. It is composed of several blue and yellow rectangular blocks arranged in a way that suggests the letter 'R'.

ANÁLISIS DE DATO CON R



contáctenos: enei@inei.gob.pe / 433-3127



Pasaje Hernán Velarde 285 Lima.

Entre la cuadra 01 y 02 de la Av. Arequipa.

Correo: enei@inei.gob.pe / campusvirtual@inei.gob.pe

Teléfonos: 433-3127 - 332-4650

Centro Andino de Formación y Capacitación en Estadística

Cursos Especializados en Estadística e Informática

**Recuerda siempre nuestro correo
para cualquier consulta**



campusvirtual@inei.gob.pe

CURSO BASICO DE “R”



Revisión

Suma de vectores

```
> x <- c(2, -3, 8)
> y <- c(1, 2, 3)
> x + y
[1] 3 -1 11
```

Combinación lineal de dos vectores

```
> 5 * x - 2 * y
[1] 8 -19 34
```

Producto y cociente de vectores

```
> x * y
[1] 2 -6 24
> x/y
[1] 2.000000 -1.500000 2.666667
```

Revisión

Covarianza y coeficiente de correlación

```
> cov(x, y)
[1] 3
> cor(x, y)
[1] 0.5447048
```

Funciones de uso general

Logaritmos neperianos de las componentes

```
> log(y)
[1] 0.0000000 0.6931472 1.0986123
```

Logaritmos en diferentes bases (por ejemplo, base 10)

```
> log(y, 10)
[1] 0.0000000 0.3010300 0.4771213
```


Revisión

Potencias de base e

```
> exp(y)
[1] 2.718282 7.389056 20.085537
```

Potencias en cualquier otra base (por ejemplo, 2)

```
> 2^y
[1] 2 4 8
```

Otras funciones

choose(m,n) da el número de combinaciones de m elementos de orden n

```
> choose(10, 2)
[1] 45
```

factorial(n) da el factorial de un numero (n), o lo que es lo mismo, calcula las permutaciones de n elementos

```
> factorial(6)
[1] 720
```

Revisión

runif(n) genera al azar n valores comprendidos entre 0 y 1

```
> runif(5)
[1] 0.55932671 0.52277705 0.60363387 0.01572491 0.90898364
```

Funciones generadas por el usuario

Ejercicio: a crear una función que a cada componente de un vector le reste la media de todas las componentes del vector:

function es la instrucción para crear una función.

La **x** que le sigue es el argumento de la función y entre llaves esta lo que la función hace con el argumento. En este ejemplo a cada componente del vector le resta la media de todas las componentes. Para ejecutar la función basta con introducir el nombre de la misma seguido del argumento entre paréntesis

Revisión

Introducción de datos

Para introducir **pocos datos**, podemos hacerlo como hasta ahora

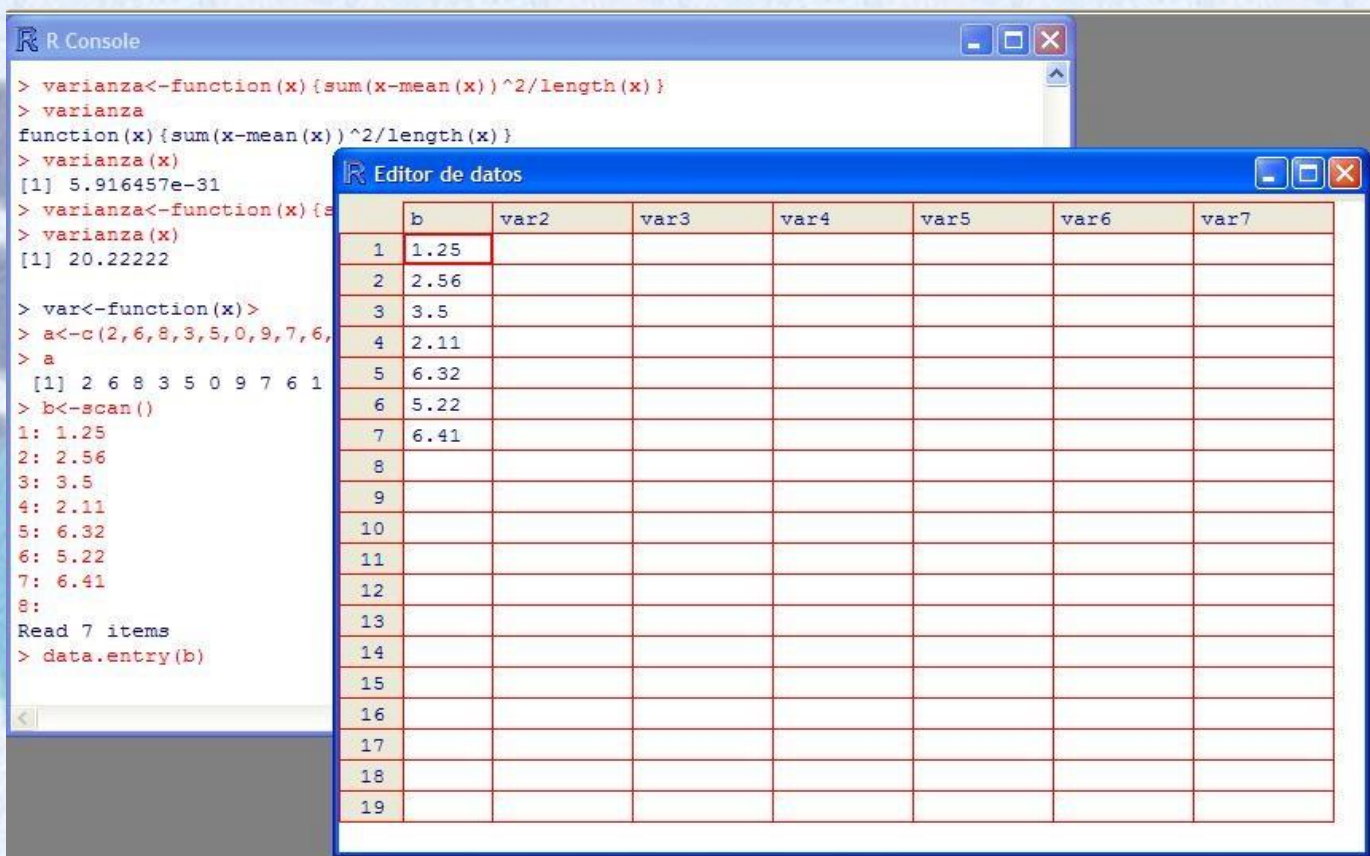
```
> a <- c(2, 6, 8, 3, 5, 0, 9, 7, 6, 1)
```

Para cantidades **no muy grandes** de datos podemos utilizar **scan()**

```
> b<-scan()  
1: 1.25  
2: 2.56  
3: 3.5  
4: 2.11  
5: 6.32  
6: 5.22  
7: 6.41  
8:  
Read 7 items
```

Al introducir la instrucción nos aparece 1 y el programa queda a la espera de que introduzcamos un valor, lo hacemos y damos a Enter y aparece 2 para que introduzcamos el siguiente valor. Así continuaremos hasta el final. Para indicar que terminamos y que ya no hay mas datos debemos hacer Enter sin introducir ningun valor. Si queremos editar una variable ya introducida podemos usar la función `data.entry(b)`, que abre una ventana como la siguiente:

Revisión



The screenshot displays two R Studio windows. The 'R Console' window on the left contains the following code and output:

```
> varianza<-function(x){sum(x-mean(x))^2/length(x)}  
> varianza  
function(x){sum(x-mean(x))^2/length(x)}  
> varianza(x)  
[1] 5.916457e-31  
> varianza<-function(x){sum(x-mean(x))^2/length(x)}  
> varianza(x)  
[1] 20.22222  
  
> var<-function(x)>  
> a<-c(2,6,8,3,5,0,9,7,6,1)  
> a  
[1] 2 6 8 3 5 0 9 7 6 1  
> b<-scan()  
1: 1.25  
2: 2.56  
3: 3.5  
4: 2.11  
5: 6.32  
6: 5.22  
7: 6.41  
8:  
Read 7 items  
> data.entry(b)
```

The 'R Editor de datos' window on the right shows a data entry table with 19 rows and 8 columns. The first column is labeled 'b' and contains the values 1.25, 2.56, 3.5, 2.11, 6.32, 5.22, and 6.41 for rows 1 through 7. The remaining columns are labeled var2 through var7 and are currently empty.

	b	var2	var3	var4	var5	var6	var7
1	1.25						
2	2.56						
3	3.5						
4	2.11						
5	6.32						
6	5.22						
7	6.41						
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							

Revisión

Podemos agregar un nuevo valor en la 8 y cerrar la ventana. De esta forma el nuevo valor quedara incorporado al vector b:

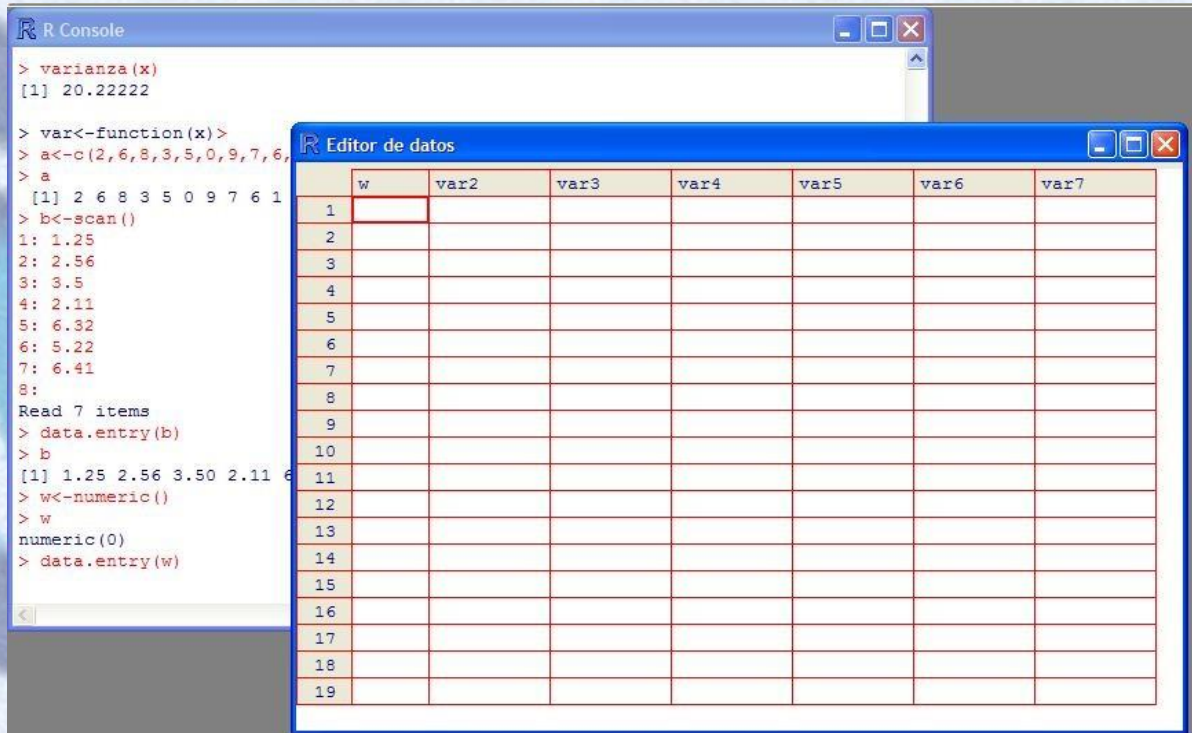
```
> data.entry(b)
> b
[1] 1.25 2.56 3.50 2.11 6.32 5.22 6.41 3.41
```

Para introducir datos en una variable nueva con data.entry(x), creamos un vector vaco. Por ejemplo:

```
> w <- numeric()
> w
numeric(0)
> data.entry(w)
```

Revisión

Se nos desplegará una ventana como la anterior, en la que la primera columna estará encabezada por el nombre del vector vaco que hemos creado. En dicha columna podremos introducir los valores que queramos.



The screenshot shows two windows from the R environment. The 'R Console' window displays the following commands and output:

```
> varianza(x)
[1] 20.22222

> var<-function(x)>
> a<-c(2,6,8,3,5,0,9,7,6,
> a
[1] 2 6 8 3 5 0 9 7 6 1
> b<-scan()
1: 1.25
2: 2.56
3: 3.5
4: 2.11
5: 6.32
6: 5.22
7: 6.41
8:
Read 7 items
> data.entry(b)
> b
[1] 1.25 2.56 3.50 2.11 6
> w<-numeric()
> w
numeric(0)
> data.entry(w)
```

The 'Editor de datos' window shows a data entry table with 19 rows and 7 columns. The first column is labeled 'w' and the other columns are labeled 'var2' through 'var7'.

	w	var2	var3	var4	var5	var6	var7
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							

Gráficos en R

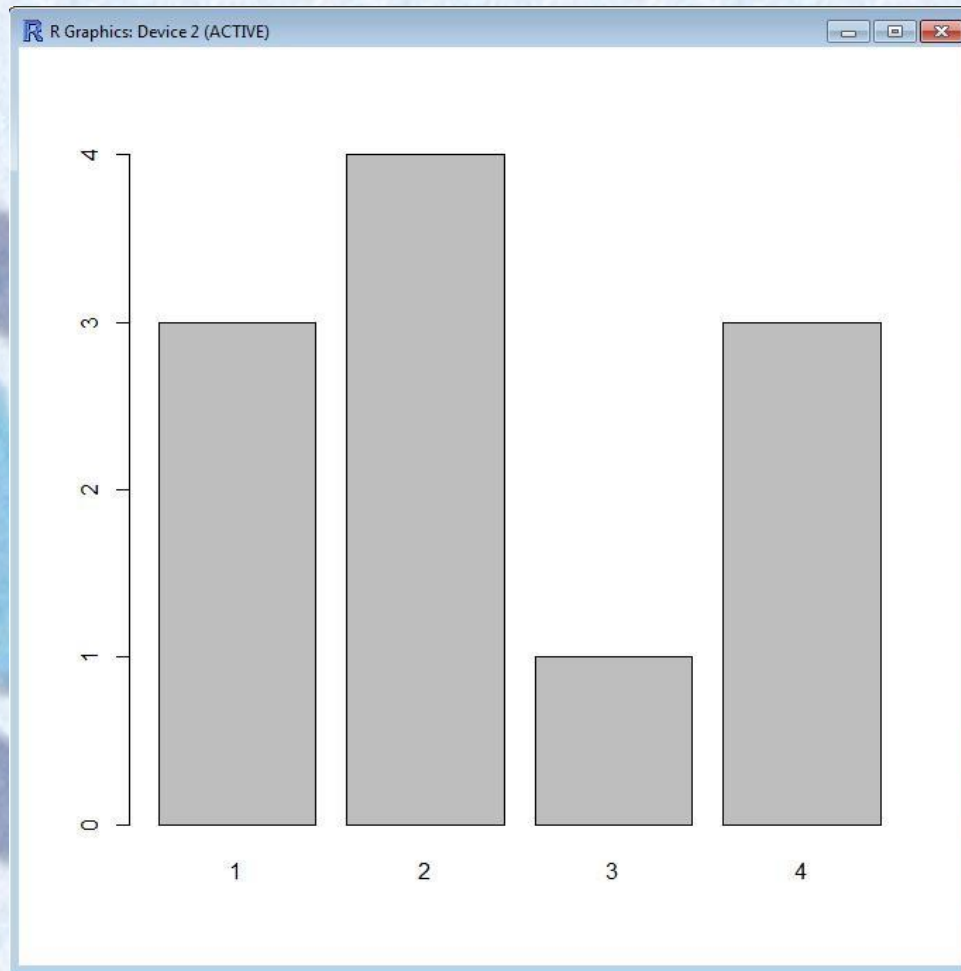
Diagrama de barras

La función **table** tabula los datos en `x` `table(x)` produce:

Este resultado nos indica que el 1 aparece tres veces, el 2 cuatro veces, el 3 una vez y el 4 tres veces. La instrucción **barplot(table(x))** muestra el diagrama de barras:

```
> table(x)
x
1 2 3 4
3 4 1 3
> barplot(table(x))
```

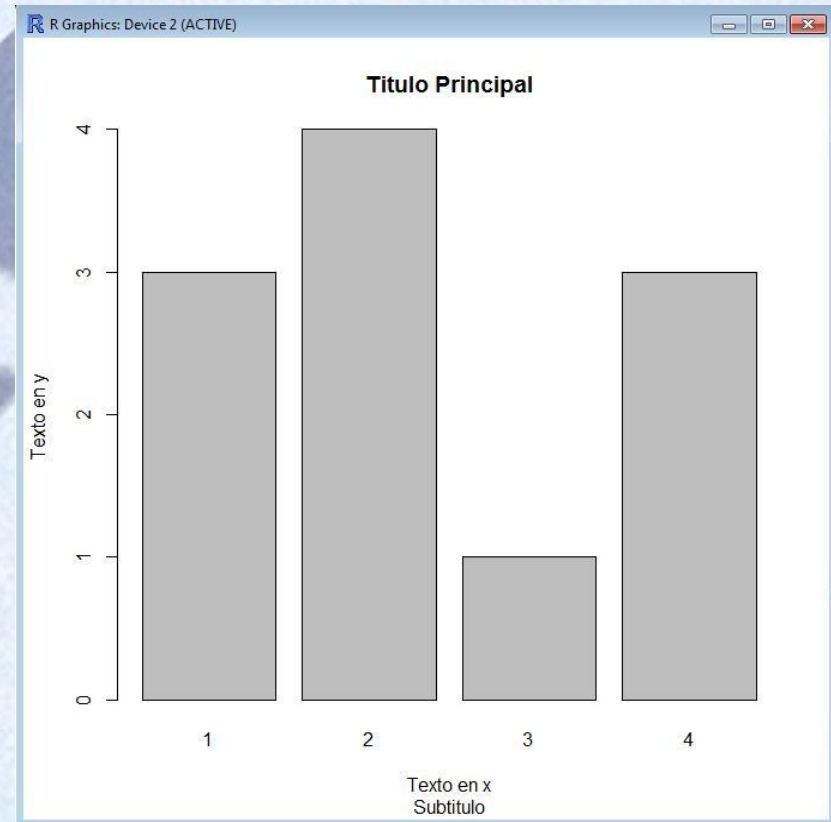
Gráficos en R



Gráficos en R

El diagrama se puede completar con títulos, subtítulos y etiquetas en los ejes.

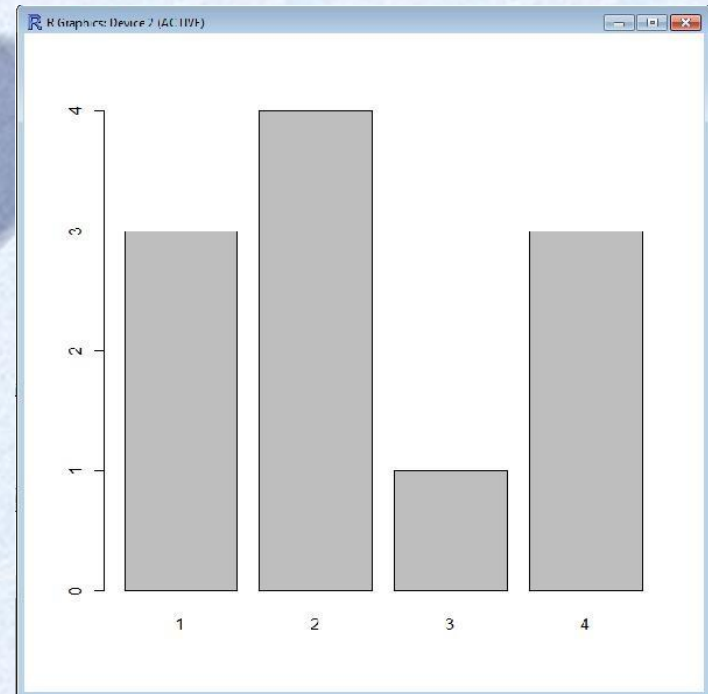
```
barplot(table(x), main = "Titulo  
principal", sub = "Subtitulo",  
xlab = "Texto en x", ylab = "Texto  
en y")
```



Gráficos en R

Si tenemos los datos en un vector x y las frecuencias correspondientes en otro vector y , podemos también obtener el diagrama de barras procediendo de la siguiente forma:

```
> x <- c(1, 2, 3, 4)  
> y <- c(3, 4, 1, 3)  
> barplot(y, names.arg = x)
```



Gráficos en R

Histograma

Consideremos una variable con 100 valores generados de la siguiente forma:

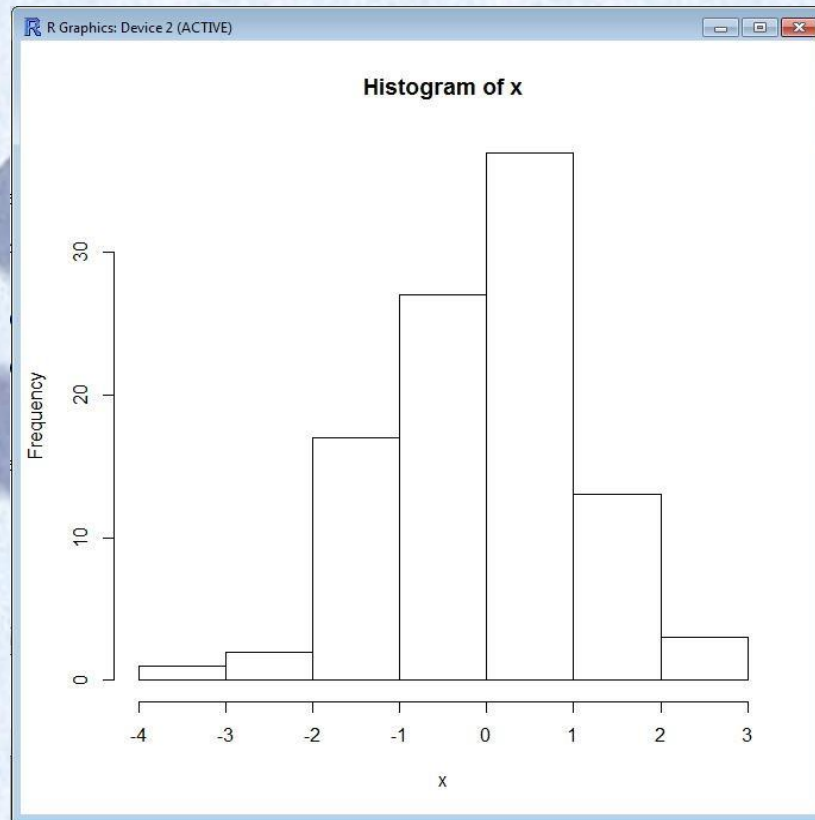
rnorm(100) genera 100 cien valores y **set.seed(111)** hace que en todos los ordenadores los numeros generados sean los mismos. Probemos:

```
> set.seed(111)
> x <- rnorm(100)
> x
[1] 0.23522071 -0.33073587 -0.31162382 -2.30234566 -0.17087604 0.14027823 -1.49742666 -1.01018842
[9] -0.94847560 -0.49396222 -0.17367413 -0.40659878 1.84563626 0.39405411 0.79752850 -1.56666536
[17] -0.08585101 -0.35913948 -1.19360897 0.36418674 0.36166245 0.34696437 0.18973653 -0.15957681
[25] 0.32654924 0.59825420 -1.84153430 2.71805560 0.19124439 -1.30129607 -3.11321730 -0.94135740
[33] 1.40025878 -1.62047003 -2.26599596 1.16299359 -0.11615504 0.33425601 -0.62085811 -1.30984491
[41] -1.17572604 -1.12121553 -1.36190448 0.48112458 0.74197163 0.02782463 0.33137971 0.64411413
[49] 2.48566156 1.95998171 0.19166338 1.55254427 0.91424229 0.35862537 0.17509564 -0.84726777
[57] 0.97823166 1.80586826 0.12291480 -0.12977203 -0.21642866 1.44647817 0.40970980 0.91091657
[65] 1.43035817 -0.38129196 0.20230718 -0.80619919 0.29463418 1.40488308 1.02376685 0.47612606
[73] -0.67033033 0.15923432 -0.38271538 0.93576259 -0.63153227 -0.09830608 1.03198498 0.38780843
[81] -1.25612931 -0.78695273 0.42981155 -0.37641622 -1.21622907 1.02927851 0.43039700 -1.24557402
[89] -0.60272849 0.66006939 2.05074953 0.49080818 -1.73147942 0.71088366 0.01382291 -1.40104160
[97] 1.25912367 0.12747752 0.72938651 1.21136136
```

Gráficos en R

El histograma se puede obtener con la instrucción `hist()`. Para obtener el numero de intervalos se usa la formula de Sturges.

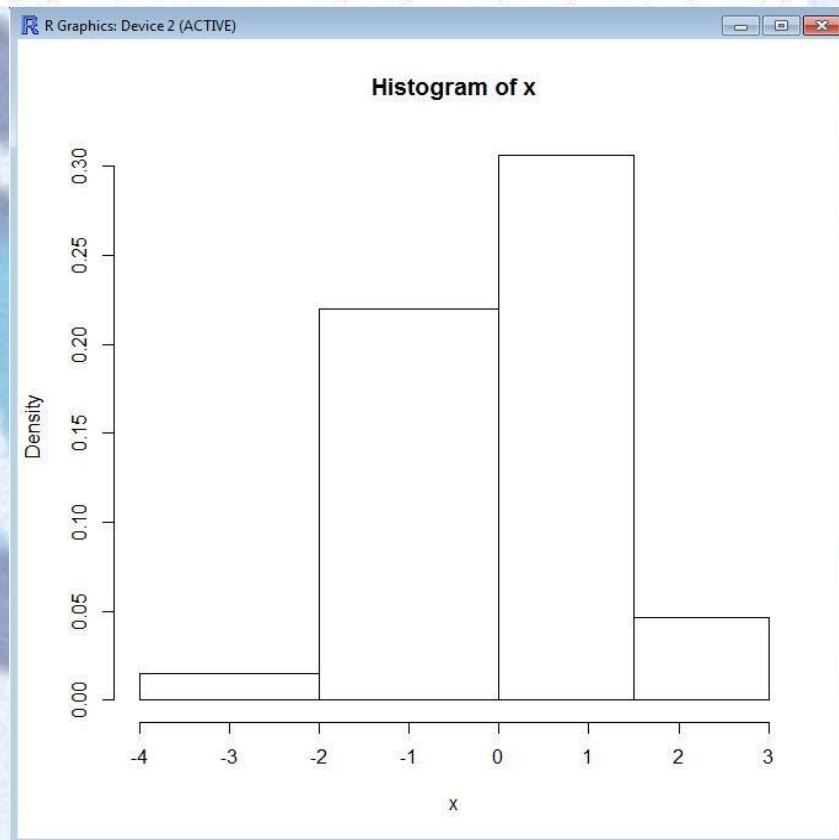
```
> hist(x)
```



Gráficos en R

Podemos elegir nuestros propios intervalos. Por ejemplo

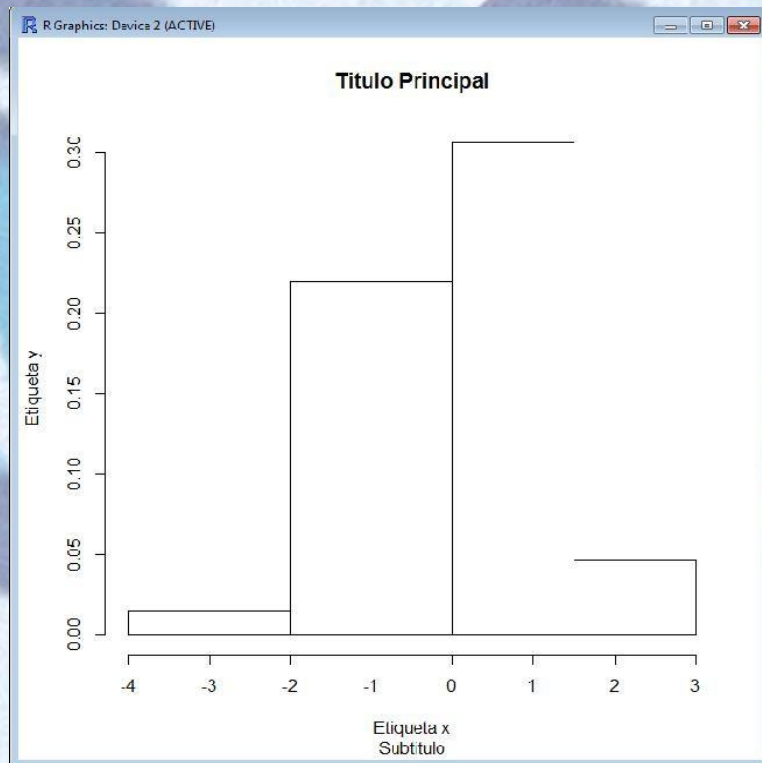
```
> hist(x, breaks = c(-4, -2, 0, 1.5, 3))
```



Gráficos en R

Se pueden añadir títulos, subtítulos, etc. de la misma forma que con barplot():

```
> hist(x, breaks = c(-4, -2, 0, 1.5, 3), main = "Titulo Principal",  
+ sub = "Subtitulo", xlab = "Etiqueta x", ylab = "Etiqueta y")
```



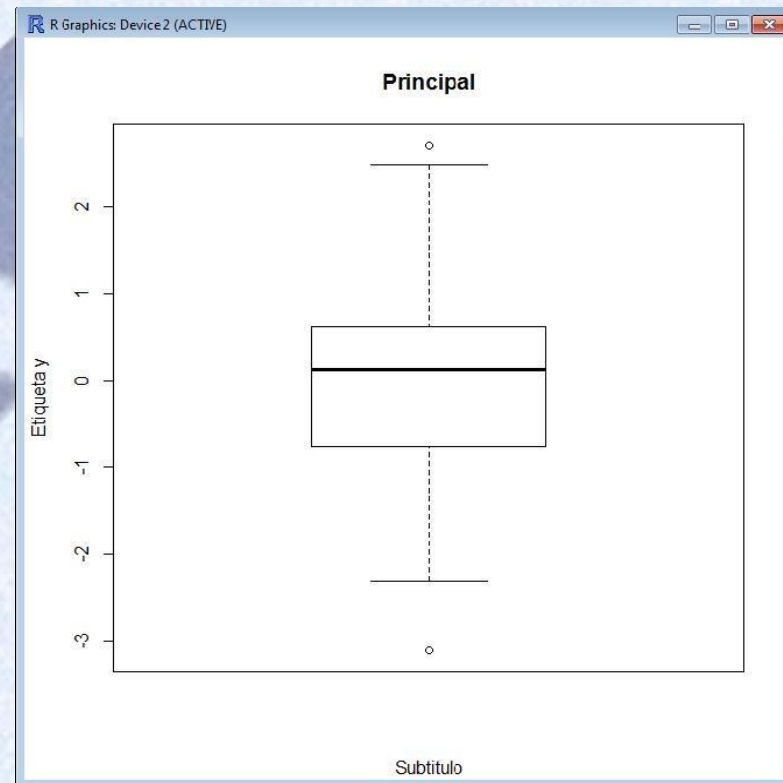
Gráficos en R

Diagramas de cajas y bigotes

Con **boxplot()** generamos diagramas de cajas y bigotes

```
boxplot(x, main = "Principal",
sub = "Subtitulo", ylab =
"Etiqueta y")
```

Podemos ver dos valores atpicos (outliers), uno por cada extremo.



Gráficos en R

Diagrama de tallos y hojas

Con las instrucciones siguientes, vamos a generar 40 valores entre 0.1 y 10.

```
> ##Diagrama de Tallos y Hojas
> set.seed(111)
> y <- ceiling(100 * runif(40))/10
> y
[1] 6.0 7.3 3.8 5.2 3.8 4.2 0.2 5.4 4.4 1.0 5.6 6.0 0.7 0.5 1.6 4.5 1.8 9.7 3.2 6.2 4.4 2.9 3.5 3.9 9.7
[26] 3.3 6.6 2.9 7.9 6.0 0.6 5.1 4.7 4.7 3.6 7.2 1.2 7.9 6.5 8.1
```


Comunicación constante con la Escuela del INEI

Correo de la Escuela del INEI
enei@inei.gob.pe

Área de Educación Virtual
campusvirtual@inei.gob.pe

