

Introducción a la Computación Científica con Python

Clase 5: SciPy

Diego Passarella

Víctor Viana

Contenido

- Ajuste de funciones no polinomiales
- Raíces de funciones
- Mínimos (optimización en 1-D)
- Nociones básicas de funciones estadísticas

Ajuste de funciones

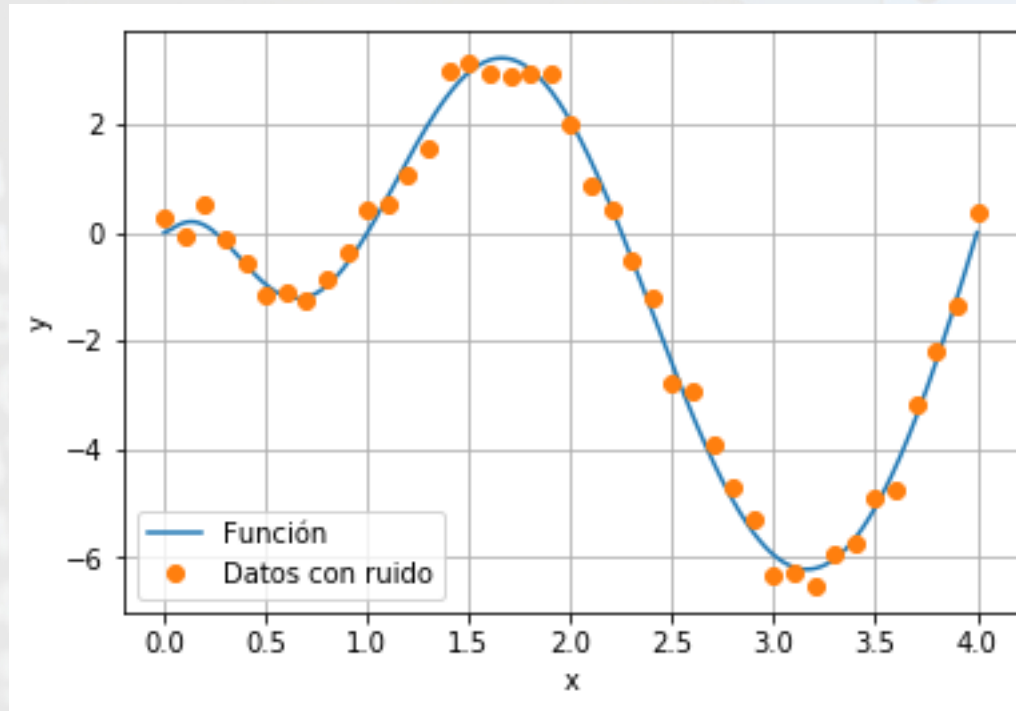
Nos basamos en la información de:

https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html

Empleo del subpaquete y sus funciones:

```
from scipy.optimize import curve_fit  
  
scipy.optimize.curve_fit(f, xdata, ydata, p0=None, ..., bounds=(-  
inf, inf, method=None, jac=None, ...))
```

Ajuste de funciones



A partir de una función conocida, generamos datos con ruido que la describen.

$$f(x) = a_0 * \sin(a_1 * x ** a_2) * x$$

$$a_0 = 2$$

$$a_1 = 2\pi$$

$$a_2 = 0.5$$

Ajuste de funciones

Distintos tipos de ajuste:

```
popt, pcov = curve_fit(FunTest2, x2, y2)
```

```
popt = [1.92291086 1.11019137 1.26125894]
```

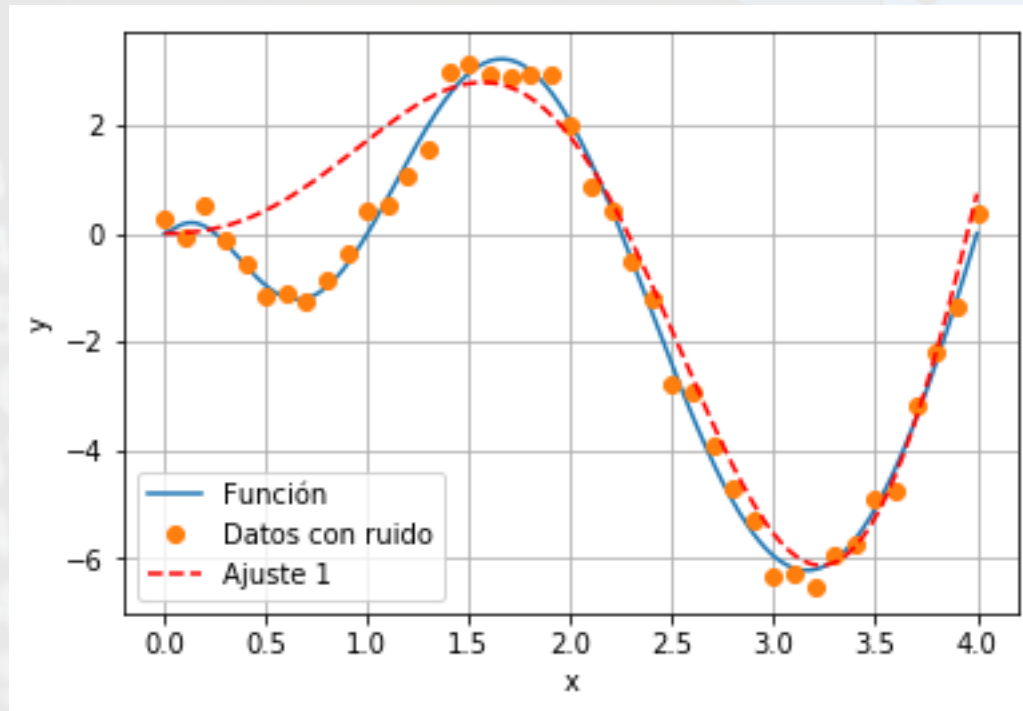
```
popt2, pcov2 = curve_fit(FunTest2, x2, y2, bounds=([0, 4, 0],  
[np.inf, np.inf, 0.8]))
```

```
popt2 = [1.99088191 6.27496869 0.50137356]
```

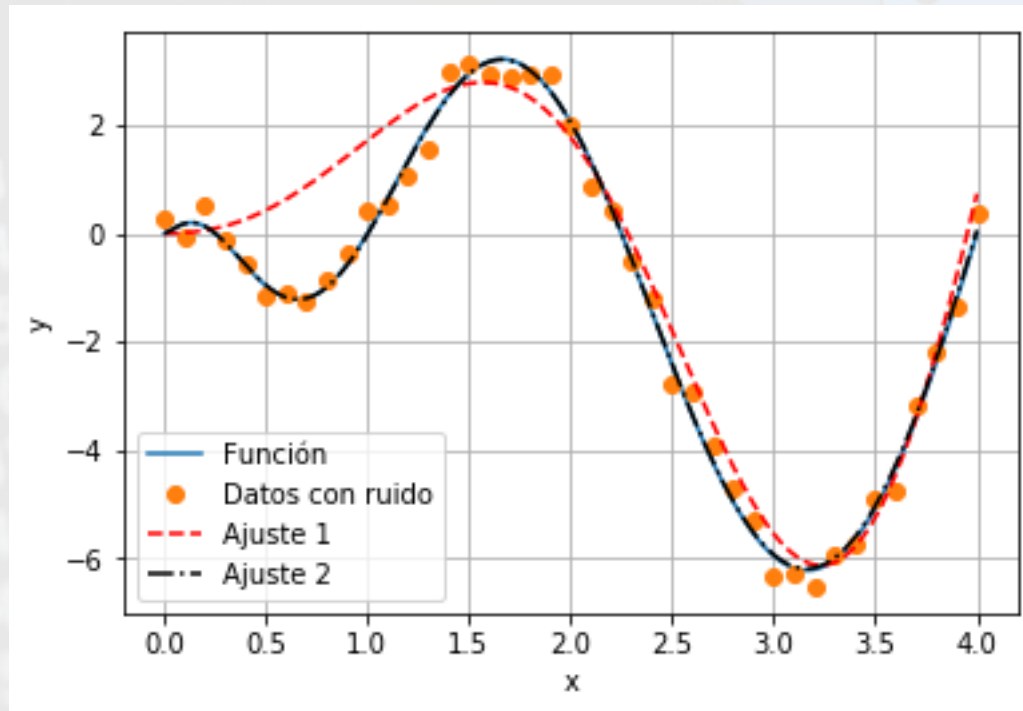
```
popt3, pcov3 = curve_fit(FunTest2, x2, y2, [1.7, 5.88, 0.39])
```

```
popt3 = [1.99088191 6.2749683 0.50137361]
```

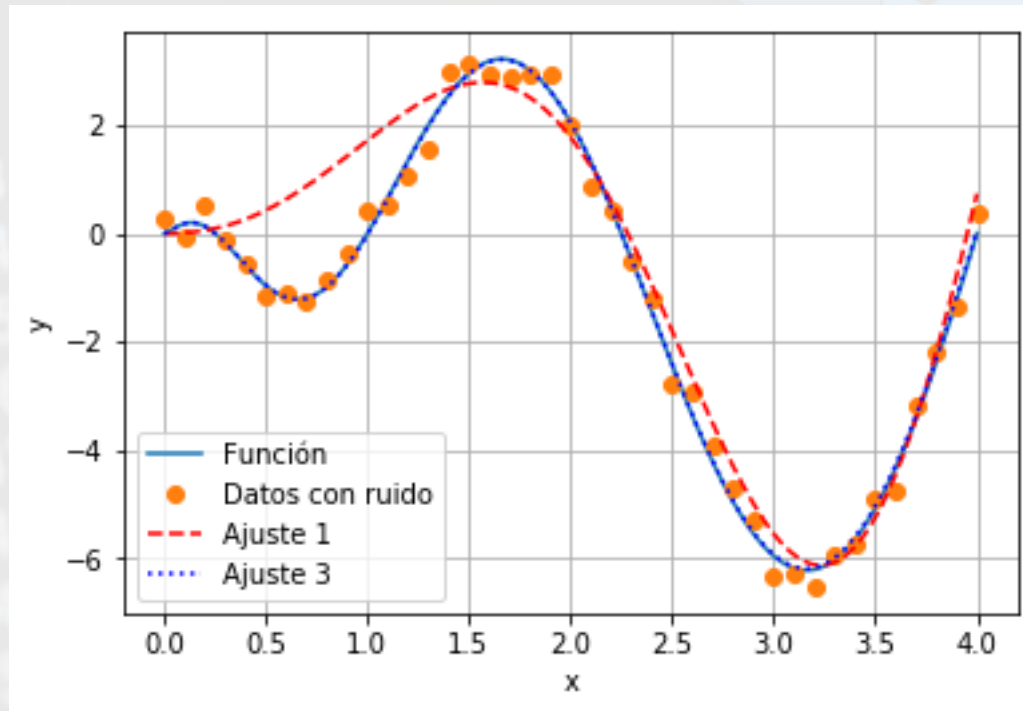
Ajuste de funciones



Ajuste de funciones



Ajuste de funciones



Raíces de funciones

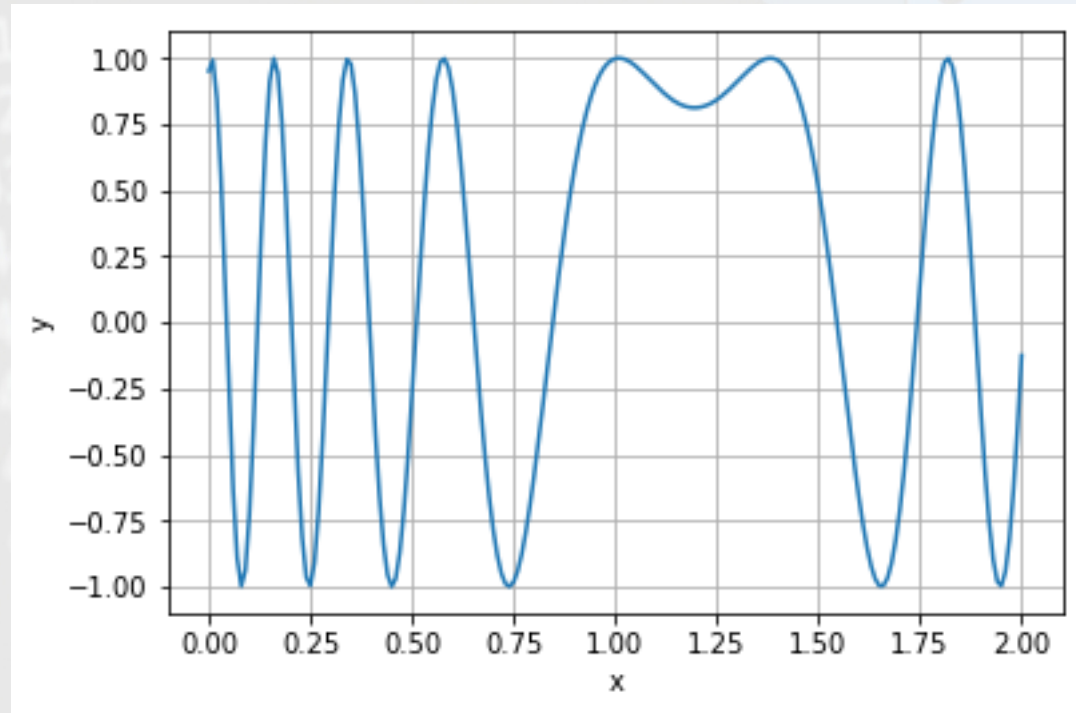
Del sub-paquete optimize:

https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.root_scalar.html

```
from scipy import optimize  
  
scipy.optimize.root_scalar(f, args=(), bracket=None, fprime=None,  
x0=None, method=None, ...)
```

Existen diversos métodos para encontrar la raíz de la función dada en el intervalo definido (bracket). Algunas involucran solamente a la función y otras también a su derivada.

Raíces de funciones



Mínimos (optimización en 1-D)

El sub-paquete optimize tiene programada una función genérica que minimiza por mínimos cuadrados una función objetivo $F(x)$, a partir de los residuales $f(x)$ y para una dada función de pérdida φ

$$\min F(x) = \frac{1}{2} \sum_i \varphi(f_i(x)^2)$$

https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html

conocer la función que los describe.

06/11/2020

```
from scipy.optimize import least_squares
```

Mínimos (optimización en 1-D)

conocer la función que los describe. Se importa la función en particular:

```
from scipy.optimize import least_squares  
  
scipy.optimize.least_squares(fun, x0, jac='2-point', bounds=(- inf,  
inf, method='trf', ...)
```

Se prueban los métodos 'trf', 'dogbox', 'lm' y ver la influencia de x0 y bounds.

Bases de funciones estadísticas

```
from scipy import stats
```

<https://docs.scipy.org/doc/scipy/reference/tutorial/stats/continuous.html>

Existen una variedad de distribuciones continuas de probabilidad en SciPy.

Los métodos más comunes son:

rvs (variable aleatoria dentro de la distribución)

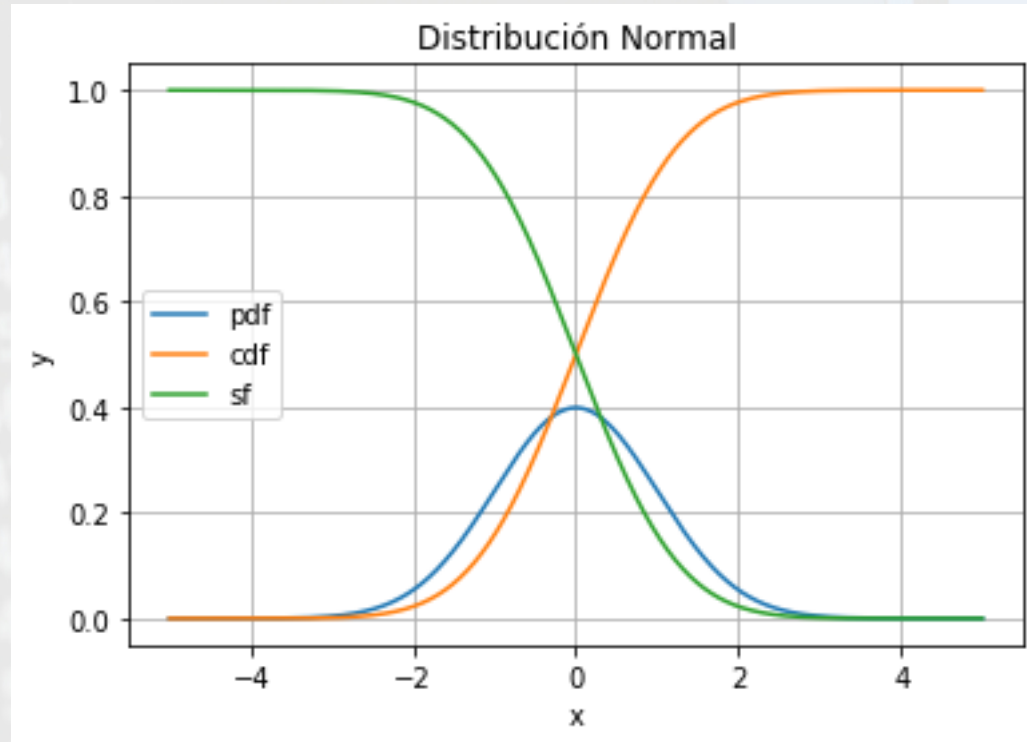
pdf (función de densidad de probabilidad)

cdf (función de probabilidad acumulada)

stats (valores característicos de los puntos analizados)

06/11/2020

Bases de funciones estadísticas



Bases de funciones estadísticas

El ejemplo de la distribución normal:

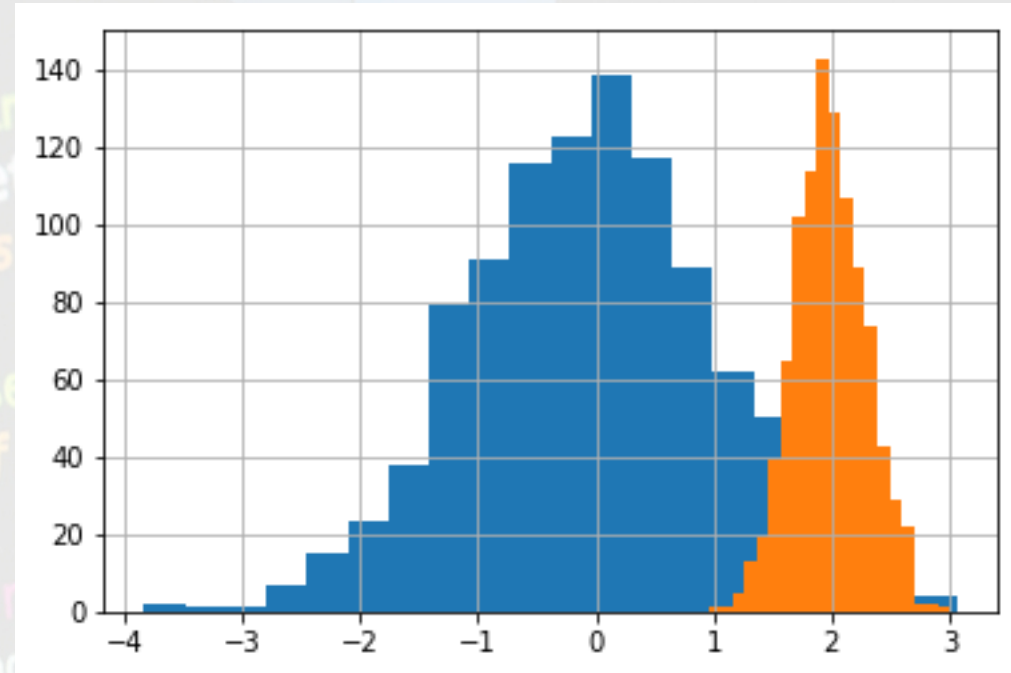
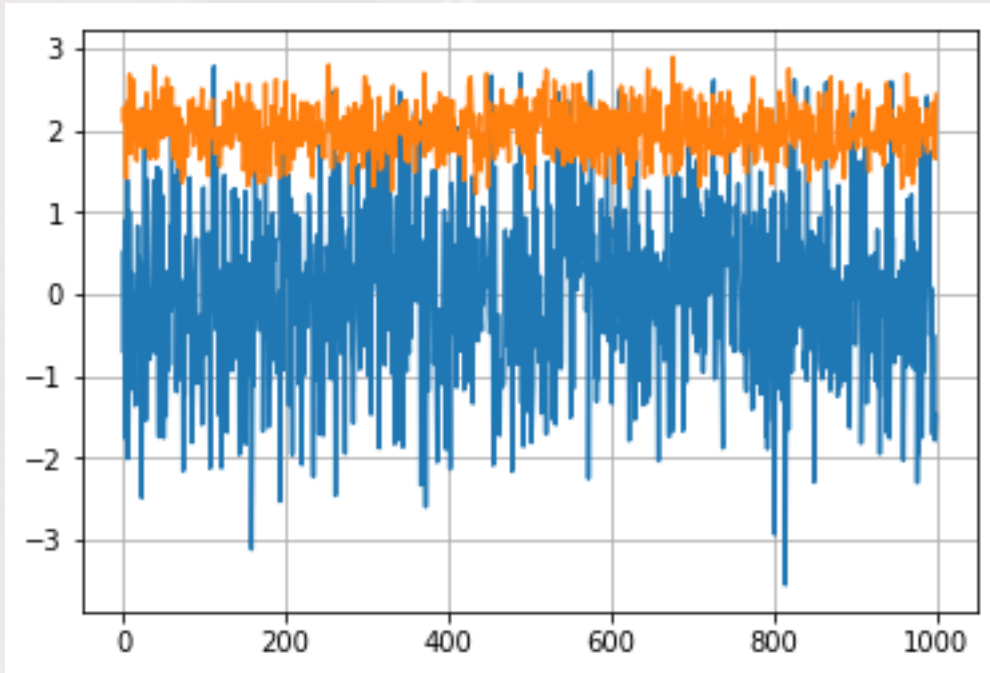
```
y = norm.rvs(size=1000) (1000 puntos distribuidos de forma normal, con media cero y varianza 1)
```

```
y2 = norm.rvs(loc = 2, scale = 0.3, size=1000) (1000 puntos distribuidos de forma normal, con media 2 y varianza 0.3)
```

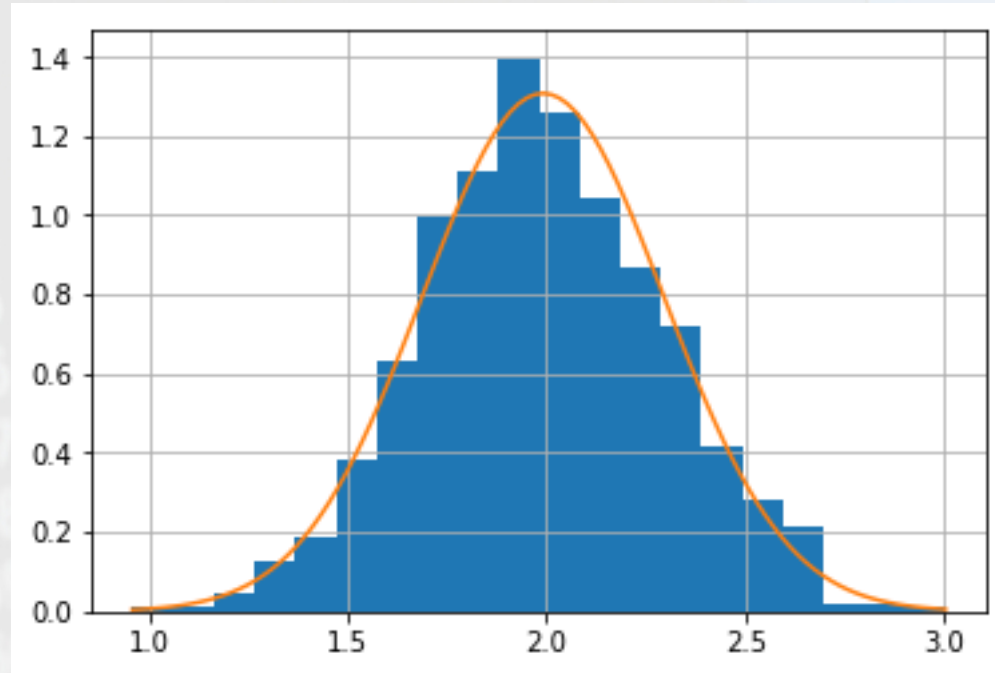
¿Cómo se hace para ajustar una distribución?

```
[media, varia] = norm.fit(y2) (esta media y varianza van a ser empleados en el reescalado de la distribución)
```

Bases de funciones estadísticas



Bases de funciones estadísticas



Bases de funciones estadísticas

