

# Tutorial Sagemath

Sagemath es practicamente una calculadora científica moderna.

## Números

```
1+3
```

4

Puede calcular fracciones

```
show(22/7)
```

$$\frac{22}{7}$$

(Como podemos ver 'show' permite imprimir todo tipo de variables en un representación más pintoresca,convirtiendo a código LaTeX la representación del cálculo)

Y su valor en decimales

```
numerical_approx(22/7)
```

3.14285714285714

Eligiendo la cantidad de decimales

```
numerical_approx(22/7, digits=10)
```

3.142857143

También puede manejar números irracionales

```
show(sqrt(2))
```

$$\sqrt{2}$$

```
show(expand((1+sqrt(2))^2))
```

$$2\sqrt{2} + 3$$

Sage tambien cuenta con constantes predefinidas

```
show(pi)
```

$\pi$



```
show(f(z))
```

$$-z^2 + 1$$

```
show(f.factor())
```

$$-(x + 1)(x - 1)$$

```
show((1/f(x)).factor())
```

$$-\frac{1}{(x + 1)(x - 1)}$$

```
show((1/f(x).factor()).partial_fraction())
```

$$\frac{1}{2(x + 1)} - \frac{1}{2(x - 1)}$$

## Resolución de ecuaciones

Sage posee herramientas que nos facilitarán la vida al momento de encontrarnos con ecuaciones

```
show(solve([2*x-1==0], x))
```

$$\left[ x = \left( \frac{1}{2} \right) \right]$$

```
var('a b c'); show(solve([a*x^2-b*x+c == 0], x))
```

$$\left[ x = \frac{b - \sqrt{b^2 - 4ac}}{2a}, x = \frac{b + \sqrt{b^2 - 4ac}}{2a} \right]$$

```
var('y d e f'); s = solve([a*x+b*y == c, d*x + e*y == f], x ,y); show(s)
```

$$\left[ \left[ x = -\frac{ce - bf}{bd - ae}, y = \frac{cd - af}{bd - ae} \right] \right]$$

```
[ss.subs(a=2,b=1, c=5, d=1, e=1, f=3).show() for ss in s[0]]
```

$$x = 2$$

$$y = 1$$

[None, None]

```
((1-x^2)/(1-x)).show()
```

$$\frac{x^2 - 1}{x - 1}$$

```
((1-x^2)/(1-x)).full_simplify().show()
```

$$x + 1$$

```
g(x) = ((sin(x)^4-cos(x)^4)/(sin(x)^2-cos(x)^2)); show(g)
```

$$x \mapsto \frac{\cos(x)^4 - \sin(x)^4}{\cos(x)^2 - \sin(x)^2}$$

```
g.full_simplify().show()
```

$$1$$

Sage tambien permite resolver sistemas de ecuaciones.

```
var('x','y')  
seq = [x+y == 10, 2*x+y == 24]  
show(seq)
```

$$[x + y = 10, 2x + y = 24]$$

```
show(solve(seq, x ,y))
```

$$[[x = 14, y = (-4)]]$$

## Sumatorias

```
var('k n'); sum(x^k, k, 0, n).show()
```

$$\frac{x^{n+1} - 1}{x - 1}$$

```
sum(binomial(n,k)*x^k, k, 0,n).show()
```

$$(x + 1)^n$$

```
sum(1/k^2, k, 1, oo).show()
```

$$\frac{1}{6} \pi^2$$

## Derivadas

```
p(x) = 3*x^2+x-5; p.show()
```

$$x \mapsto 3x^2 + x - 5$$

```
derivative(p,x).show()
```

$$x \mapsto 6x + 1$$

```
derivative(p(x), x, 2).show()
```

6

```
ee(x) = exp(-x^2); show(ee)
```

$$x \mapsto e^{-x^2}$$

```
ee.derivative().show()
```

$$x \mapsto -2xe^{-x^2}$$

## Integrales

```
integral(x^2, x).show()
```

$$\frac{1}{3}x^3$$

```
integral(x^2, x, 0, 1).show()
```

$$\frac{1}{3}$$

```
integral(sin(x), x).show()
```

$$-\cos(x)$$

```
integral(1/(x+1), x).show()
```

$$\log(x + 1)$$

## Vectores y Matrices

Podemos crear matrices al estilo Python (lista de listas) y utilizar funcionalidades de Sage para hacer álgebra sobre estas.

```
var('x', 'y')  
V = vector([x,y,1])  
show(V)
```

$$(x, y, 1)$$

```
M = matrix([[2, -2, 1], [-1, 3, -1], [2, -4, 3]])  
show(M)
```

$$\begin{pmatrix} 2 & -2 & 1 \\ -1 & 3 & -1 \\ 2 & -4 & 3 \end{pmatrix}$$

```
show(V*M)
```

$$(2x - y + 2, -2x + 3y - 4, x - y + 3)$$

```
show(M.transpose())
```

$$\begin{pmatrix} 2 & -1 & 2 \\ -2 & 3 & -4 \\ 1 & -1 & 3 \end{pmatrix}$$

```
show(M.inverse())
```

$$\begin{pmatrix} \frac{5}{6} & \frac{1}{3} & -\frac{1}{6} \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ -\frac{1}{3} & \frac{2}{3} & \frac{2}{3} \end{pmatrix}$$

También podemos obtener los vectores y valores propios de una matriz.

```
EV = M.eigenvectors_left()  
show(EV)
```

$$\left[ (6, [(1, -2, 1)], 1), \left( 1, \left[ \left( 1, 0, -\frac{1}{2} \right), \left( 0, 1, \frac{1}{2} \right) \right], 2 \right) \right]$$

```
show(EV[1])
```

$$\left( 1, \left[ \left( 1, 0, -\frac{1}{2} \right), \left( 0, 1, \frac{1}{2} \right) \right], 2 \right)$$

Esto nos dice que 1 es un valor propio de M, con 2 vectores propios asociados, y por lo tanto 2 de multiplicidad (representado por el 2 final).

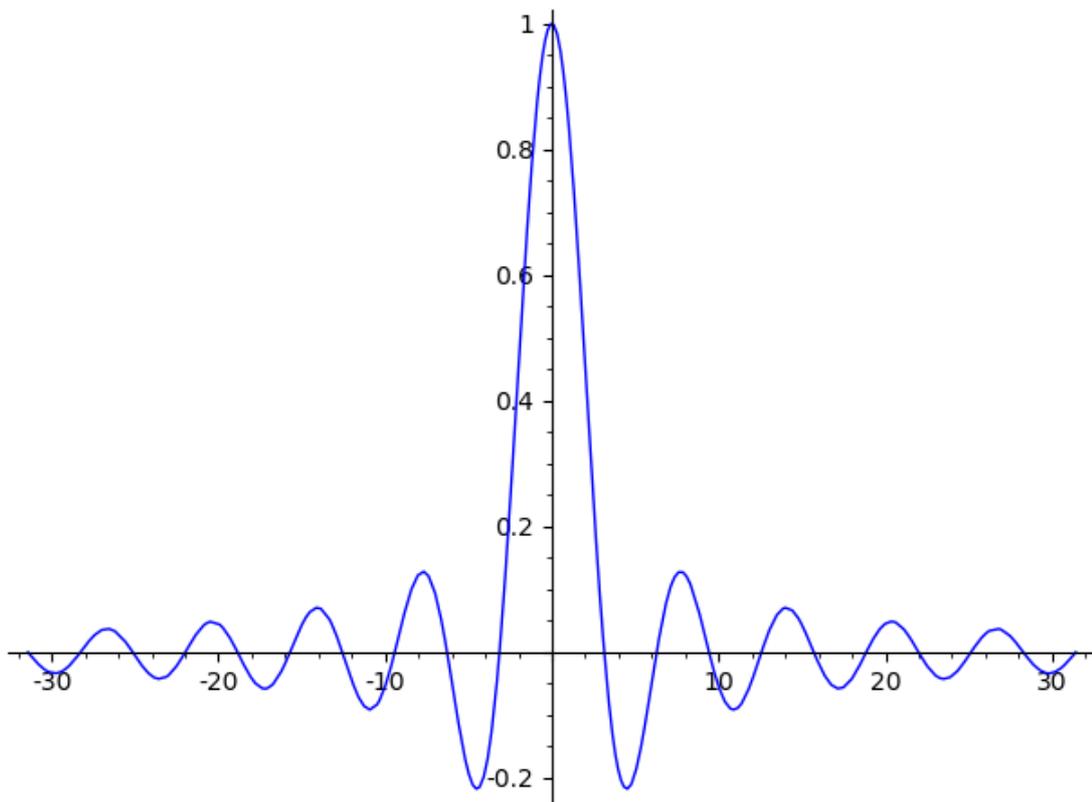
## Gráficos y límite sobre funciones

Sage puede graficar fácilmente funciones en un rango determinado

```
f(x) = sin(x)/x; show(f)
```

$$x \mapsto \frac{\sin(x)}{x}$$

```
plot(f, (-10*pi, 10*pi))
```



Debido a que esta función no está definida en  $x=0$ , no podemos evaluar en ese punto, pero sí encontrar el límite

```
limit(f(x), x=0)
```

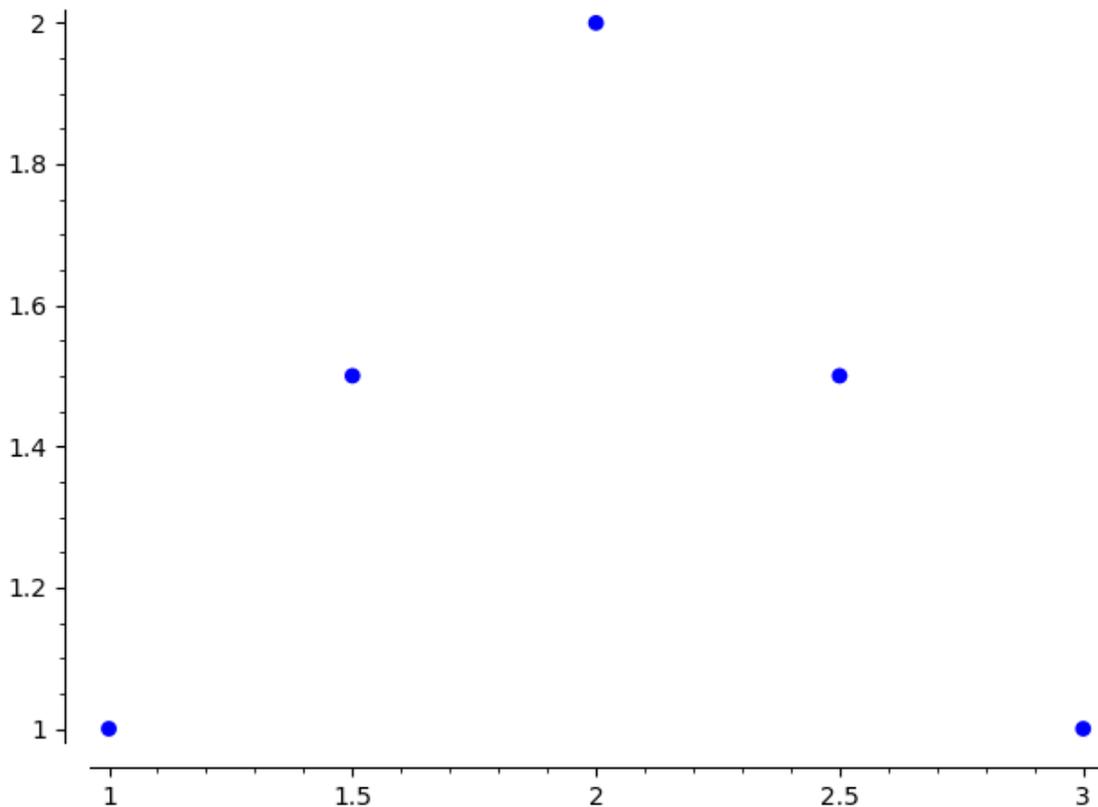
1

Sage utiliza Matplotlib para manejar gráficos de todo tipo. Con esta librería podemos tomar líneas, puntos, gráficos, etc. como objetos gráficos y manejarlos a nuestra voluntad. Graphics es el elemento gráfico que almacenará nuestro gráfico.

Supongamos que tenemos una lista de tuplas (x,y) y deseamos hacer un scatterplot de ellas.

```
L = [(1,1), (1.5, 1.5), (2,2), (2.5, 1.5), (3,1)]
```

```
list_plot(L, size=40)
```



Esto fue fácil, ya que `list_plot` es una función de gráficos genérica que hará todo el trabajo por nosotros.

Però ahora queremos unir estos puntos con líneas, para esto necesitamos hacer un poco más de trabajo, debemos crear líneas con tuplas de coordenadas y luego agregarlas a nuestro Gráfico. Primero que todo debemos crear nuestro objeto que almacenará todos los gráficos, con él agregaremos nuestro anterior gráfico de puntos.

```
g = Graphics()
g += list_plot(L, size=40)
```

Para crear las líneas procedemos a dibujarlas tomando cada par consecutivo de coordenadas desde `L`.

```
for i in range(0, len(L)-1):
    g += line((L[i], L[i+1]))
```

Finalmente procedemos a visualizar nuestro gráfico.

show(g)

