

Utilisation du package Sympy

```
install.packages("rSymPy")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'  
## (as 'lib' is unspecified)
```

```
library(rSymPy)
```

```
## Loading required package: rJython
```

```
## Loading required package: rJava
```

```
## Loading required package: rjson
```

```
# On declare notre variable x pour rendre la suite possible
```

```
x <- Var("x")
```

```
# Simplification des nombres decimaux avec la fonction nsimplify
```

```
sympy("nsimplify(3.75)")
```

```
## [1] "15/4"
```

```
# Reconnaissance de nombre. R peut reconnaitre des nombres tel que pi sans donner sa valeur exacte
```

```
sympy("simplify(cos(pi/6))")
```

```
## [1] "3**(1/2)/2"
```

```
# Simplification d'expression. SymPy permet aussi de simplifier les expressions permettant  
# de passer d'une forme a une autre.
```

```
sympy("sqrt(15)")
```

```
## [1] "15**(1/2)"
```

```
# Resolution de l'equation  $x^2 - 2 = 0$ 
```

```
sympy("solve(x**2 - 2,x)")
```

```
## [1] "[2**(1/2), -2**(1/2)]"
```

```
# Derivee premiere
```

```
sympy("diff(x**3,x,1)")
```

```
## [1] "3*x**2"
```

```
# Derivee seconde  
sympy("diff(x**3,x,2)")
```

```
## [1] "6*x"
```

```
# Derivee troisieme  
sympy("diff(x**3,x,3)")
```

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

```
# SymPy peut aussi integrer des fonctions  
sympy("integrate(2*x)")
```

```
## [1] "x**2"
```

```
# Developper (x-2)*(x-2)  
sympy("((x-2)*(x-2))")
```

```
## [1] "(2 - x)**2"
```

```
sympy("(2 * (x-2))")
```

```
## [1] "-4 + 2*x"
```

#Bibliographie

<http://www.di.fc.ul.pt/~jpn/r/symbolic/>

<https://docs.sympy.org/latest/tutorial/>

<https://cran.r-project.org/web/packages/rSymPy/rSymPy.pdf>