

# Rsympy

Wenjun ZHAO

12/9/2020

Nous allons maintenant aborder les fonctions mathématiques de RStudio. Nous allons voir la fonction *rSymPy*.

Cette fonction sert à traiter les nombres pairs et impairs, les nombres réels, les complexes, les nombres premiers et l'infini. Pour commencer, nous allons installer la fonction *rSymPy*:

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

La *sympy* fonction passe une chaîne d'entrée à SymPy et renvoie la sortie. La première fois qu'il *sympy* est invoqué dans une session, il démarre également SymPy en invoquant *sympyStart* (qui définit les chemins appropriés, les appels *jythonStart* puis les importations *sympy*). En conséquence, on peut s'attendre à ce que la première invocation de *sympy* soit beaucoup plus lente que les suivantes. *jythonStart* crée une variable *.Jython* qui est stockée dans l'environnement global contenant les informations de connexion à la session SymPy / Jython.

```
install.packages("rJython")
```

```
install.packages("rJava")
```

En interne, si l'argument *output=TRUE*, la chaîne de caractères d'entrée par défaut, est précédé de *\*\_\_Rsympy=\** donc si une telle préface provoque une erreur, assurez-vous que l'argument *output=FALSE*.

Maintenant, nous allons voir certaines possibilités de base de cette librairie:

```
library(rSymPy)
```

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

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

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

```
# create a SymPy variable called x
sympy("var('x')")
```

```
## [1] "x"
```

```
sympy("y = x*x")
```

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

```
sympy("y")
```

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

```
sympy("limit(1/x, x, oo)")
```

```
## [1] "0"
```

```
# the next line fails under jython even without R  
# and seems to corrupt the rest of the session  
# sympy("(1/cos(x)).series(x, 0, 10)")
```

```
sympy("diff(sin(2*x), x, 1)")
```

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

```
sympy("diff(sin(2*x), x, 2)")
```

```
## [1] "-4*sin(2*x)"
```

```
sympy("integrate(exp(-x), (x, 0, oo))")
```

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

```
sympy("xr = Symbol('xr', real=True)")
```

```
## [1] "xr"
```

```
sympy("exp(I*xr).expand(complex=True)")
```

```
## [1] "I*sin(xr) + cos(xr)"
```

```
# Matrices are stored row by row (unlike R matrices)  
cat(sympy("A = Matrix([[1,x], [y,1]])"), "\n")
```

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

```
cat(sympy("A**2"), "\n")
```

```
## [1 + x**3, 2*x]  
## [ 2*x**2, 1 + x**3]
```