Symbolic Computation in R

João Neto

October 2014

- Recognizing numbers:
- Euler's formula

Herein we use package `rSymPy` that needs Python and Java instalattion (this library uses SymPy via Jython). If necessary cf. troubleshooting (http://stackoverflow.com/questions/2399027/cannot-load-rjava-because-cannot-load-a-shared-library).

Refs:

- Reference manual (very short) http://cran.r-project.org/web/packages/rSymPy/rSymPy.pdf (http://cran.r-project.org/web/packages/rSymPy/rSymPy.pdf)
- SymPy Tutorial: http://docs.sympy.org/latest/tutorial/ (http://docs.sympy.org/latest/tutorial/)

```
library(rSymPy)
```

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

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

```
x*x/x
```

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

```
y <- Var("x**3")
x/y
```

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

```
z <- sympy("2.5*x**2")
z + y
```

```
## [1] "2.5*x**2 + x**3"
```

```
sympy("sqrt(8).evalf()")  # evaluate an expression
```

```
## [1] "2.82842712474619"
```

```
sympy("sqrt(8).evalf(50)")
```

```
## [1] "2.8284271247461900976033774484193961571393437507539"
```

```
sympy("pi.evalf(120)")
```

```
## [1] "3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706798214808651328230665"
```

```
sympy("one = cos(1)**2 + sin(1)**2")
```

```
## [1] "cos(1)**2 + sin(1)**2"
```

```
sympy("(one - 1).evalf()")  # rounding errors
```

```
## [1] "-.0e-124"
```

```
sympy("(one - 1).evalf(chop=True)")  # rouding this type of roundoff errors
```

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

```
sympy("Eq(x**2+2*x+1,(x+1)**2)") # create an equation
```

```
## [1] "1 + 2*x + x**2 == (1 + x)**2"
```

```
sympy("a = x**2+2*x+1")
```

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

```
sympy("b = (x+1)**2")
```

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

```
"0" == sympy("simplify(a-b)")  # if they are equal, the result is zero
```

```
## [1] TRUE
```

```
# simplify works in other tasks:
sympy("simplify((x**3 + x**2 - x - 1)/(x**2 + 2*x + 1))")
```

```
## [1] "-1 + x"
```

```
sympy("(x + 2)*(x - 3)")
```

```
## [1] "-(2 + x)*(3 - x)"
```

```
sympy("expand((x + 2)*(x - 3))")
```

```
## [1] "-6 - x + x**2"
```

```
sympy("factor(x**3 - x**2 + x - 1)")
```

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

```
y <- Var("y")
z <- Var("z")
sympy("collect(x*y + x - 3 + 2*x**2 - z*x**2 + x**3, x)")  # organize equation around var 'x'
```

```
## [1] "-3 + x*(1 + y) + x**2*(2 - z) + x**3"
```

```
sympy("(x*y**2 - 2*x*y*z + x*z**2 + y**2 - 2*y*z + z**2)/(x**2 - 1)")
```

```
## [1] "-(-2*y*z - 2*x*y*z + y**2 + z**2 + x*y**2 + x*z**2)/(1 - x**2)"
```

```
sympy("cancel((x*y**2 - 2*x*y*z + x*z**2 + y**2 - 2*y*z + z**2)/(x**2 - 1))")
```

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

```
sympy("expand_func(gamma(x + 3))")
```

```
## [1] "x*(1 + x)*(2 + x)*gamma(x)"
```

```
sympy("y = x*x") # create a variable 'y' in the SymPy persistant state
```

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

```
sympy("A = Matrix([[1,x], [y,1]])")
```

```
## [1] "[   1, x]\n[x**2, 1]"
```

```
sympy("A**2")
```

```
## [1] "[1 + x**3,      2*x]\n[  2*x**2, 1 + x**3]"
```

```
sympy("B = A.subs(x,1.1)")  # replace x by 1.1 (btw, SymPy objects are immutable)
```

```
## [1] "[   1, 1.1]\n[1.21,   1]"
```

```
sympy("B**2")
```

```
## [1] "[2.331,   2.2]\n[ 2.42, 2.331]"
```

```
# more replacement, a subexpression by another:
sympy("expr = sin(2*x) + cos(2*x)")
```

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

```
sympy("expr.subs(sin(2*x), 2*sin(x)*cos(x))")
```

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

```
sympy("expr.subs(x,pi/2)")
```

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

```
# more matrix stuff:
a1 <- Var("a1")
a2 <- Var("a2")
a3 <- Var("a3")
a4 <- Var("a4")

A <- Matrix(List(a1, a2), List(a3, a4))
#define inverse and determinant
Inv <- function(x) Sym("(", x, ").inv()")
Det <- function(x) Sym("(", x, ").det()")

A
```

```
## [1] "[a1, a2]\n[a3, a4]"
```

```
cat(A,"\n")
```

```
## ( Matrix( ( [ a1,a2 ] ),( [ a3,a4 ] ) ) )
```

```
Inv(A)
```

```
## [1] "[1/a1 + a2*a3/(a1**2*(a4 - a2*a3/a1)), -a2/(a1*(a4 - a2*a3/a1))]\n[            -a3/(a1*(a4 - a2*a3/a1)),       1/(a
4 - a2*a3/a1)]"
```

```
Det(A)
```

```
## [1] "a1*a4 - a2*a3"
```

```
# create function exponential
Exp <- function(x) Sym("exp(", x, ")")
Exp(-x) * Exp(x)
```

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

```
y <- Var("y")
sympy("sqrt(8)")             # simplify expression
```

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

```
sympy("solve(x**2 - 2, x)")  # solve x^2-2=0
```

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

```
sympy("limit(1/x, x, oo)")   # limit eg, x -> Inf
```

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

```
sympy("limit(1/x, x, 0)")
```

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

```
sympy("integrate(exp(-x))")              # indefinite integral
```

```
## [1] "-exp(-x)"
```

```
sympy("integrate(exp(-x*y),x)")          # indefinite integral
```

```
## [1] "-exp(-x*y)/y"
```

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

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

```
integrate( function(x) exp(-x), 0, Inf)  # integration is possible in R
```

```
## 1 with absolute error < 5.7e-05
```

```
sympy("integrate(x**2 - y, (x, -5, 5), (y, -pi, pi))")  # definite integral
```

```
## [1] "500*pi/3"
```

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

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

```
D( expression(sin(2*x)), "x" ) # also possible in base R
```

```
## cos(2 * x) * 2
```

```
sympy("diff(sin(2*x), x, 2)")  # second derivative
```

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

```
sympy("diff(sin(2*x), x, 3)")  # third  derivative
```

```
## [1] "-8*cos(2*x)"
```

```
sympy("diff(exp(x*y*z), x, y, y)") # d^3/dxdy^2
```

```
## [1] "z*exp(x*y*z) + x*y*z**2*exp(x*y*z)"
```

```
sympy("diff(exp(x*y*z), x, z, 3)") # d^4/dxdz^3
```

```
## [1] "y*exp(x*y*z) + x*z*y**2*exp(x*y*z)"
```

```
sympy("(1/cos(x)).series(x, 0, 10)")  # taylor expansion
```

```
## [1] "1 + x**2/2 + 5*x**4/24 + 61*x**6/720 + 277*x**8/8064 + O(x**10)"
```

```
sympy("exp(x).series(x, 0, 5)")       # taylor expansion
```

```
## [1] "1 + x + x**2/2 + x**3/6 + x**4/24 + O(x**5)"
```

```
sympy("exp(x).series(x, 0, 5).removeO()")
```

```
## [1] "1 + x + x**2/2 + x**3/6 + x**4/24"
```

```
sympy("Matrix([[1, 2], [2, 2]]).eigenvals()")  # get eigenvalues of matrix
```

```
## [1] "{3/2 - 17**(1/2)/2: 1, 3/2 + 17**(1/2)/2: 1}"
```

```
sympy("latex(Integral(cos(x)**2, (x, 0, pi)))")
```

```
## [1] "$\\int_{0}^{\\pi} \\operatorname{cos}^{2}\\left(x\\right)\\,dx$"
```

This can be used within the markup text $\int_0^{\pi} \cos^2(x)\,dx$ = $\frac{1}{2}\pi$

Recognizing numbers:

`nsimplify` takes a floating point number and tries to simplify it:

```
sympy("nsimplify(4.242640687119286)")
```

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

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

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

```
sympy("nsimplify(6.28, [pi], tolerance=0.01)")
```

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

```
sympy("nsimplify(pi, tolerance=1e-5)")
```

```
## [1] "355/113"
```

```
sympy("nsimplify(pi, tolerance=1e-6)")
```

```
## [1] "51/196 + 318945**(1/2)/196"
```

```
sympy("nsimplify(29.60881, constants=[pi,E], tolerance=1e-5)")
```

```
## [1] "117/8 + 14369**(1/2)/8"
```

Euler’s formula

As seen above, `sympy` allows for the Taylor’s expansion.

To check on Euler’s formula

$$e^{i\theta} = cos\,\theta + i\;sin\,\theta$$

let's define Taylor series for $sin, cos, e^{i\theta}$,

```
theta <- Var("theta")

sin.series <- function(n) sympy(paste0(  "sin(theta).series(theta, 0, ", n, ")"))
cos.series <- function(n) sympy(paste0(  "cos(theta).series(theta, 0, ", n, ")"))
exp.series <- function(n) sympy(paste0("exp(I*theta).series(theta, 0, ", n, ")"))
```

We can see similarities from these expansions:

$sin\ \theta$ = $\theta - \frac{1}{6}\theta^3 + \frac{1}{120}\theta^5 - \frac{1}{5040}\theta^7 + \mathcal{O}(\theta^8)$

$cos\ \theta$ = $1 - \frac{1}{2}\theta^2 + \frac{1}{24}\theta^4 - \frac{1}{720}\theta^6 + \mathcal{O}(\theta^8)$

$e^{i\theta}$ = $1 + \imath\theta - \frac{1}{2}\theta^2 - \frac{1}{6}\imath\theta^3 + \frac{1}{24}\theta^4 + \frac{1}{120}\imath\theta^5 - \frac{1}{720}\theta^6 - \frac{1}{5040}\imath\theta^7 + \mathcal{O}(\theta^8)$

let's separate the terms according to i,

$e^{i\theta}$ = $1 + \imath\left(\theta - \frac{1}{6}\theta^3 + \frac{1}{120}\theta^5 - \frac{1}{5040}\theta^7\right) - \frac{1}{2}\theta^2 + \frac{1}{24}\theta^4 - \frac{1}{720}\theta^6 + \mathcal{O}(\theta^8)$

and we see that it seems to follow

$$e^{i\theta} = cos\ \theta + i\ sin\ \theta$$

We can check it by subtracting both expressions and confirm that all terms cancel:

```
n <- 50

# expansion(sin)*i
isin.series <-
  function(n) sympy(paste0("expand(I*sin(theta).series(theta, 0, ", n, "))"))

sympy( paste0( "nsimplify(",
               exp.series(n),
               "-(",
               cos.series(n),"+", isin.series(n),
               "))"
              )
     )
```

```
## [1] "O(theta**50)"
```

$\mathcal{O}(\theta^{50})$