Package 'rSymPy'

February 2, 2019

Version 0.2-1.2

Date 2010-07-31

Title R Interface to SymPy Computer Algebra System

Author G Grothendieck (SymPy itself is by Ondrej Certik and others),
Contributors: Carlos J. Gil Bellosta

Maintainer G Grothendieck <ggrothendieck@gmail.com>

Description Access SymPy computer algebra system from R via Jython.

Depends rJython

License GPL

URL http://rsympy.googlecode.com

Repository CRAN

Date/Publication 2019-02-02 13:14:04 UTC

NeedsCompilation no

R topics documented:

Sym   *sympy variables*

Description

Create and manipulate sympy variables.

Usage

```
Sym(..., retclass = c("Sym", "character"))
```

1

Arguments

`...`	Variable name as a string by which sympy should identify this varible
`retclass`	Class of object to be returned.

Details

An object of class "Sym" is internally a sympy character string. One can combine such objects using the Math and Ops R operators (see help(Math) and help(Ops) for a list). Also the following are supported: `as.character.Sym`, `as.expression.Sym`, `Ops.Sym`, `Math.Sym`, `print.Sym`, `deriv.Sym`, `Limit`, `Var.Sym`, `solve.Sym`, `Integrate`, `t.Sym`, `List`, `Matrix`, `Var`, `Zero`, `Zeros` and `Eye`.

Value

As in `retclass`.

See Also

[sympy](#)

Examples

```
## Not run:
x <- Var("x")
x+x

## End(Not run)
```

---

sympy	*sympy*

Description

Interface to the sympy computer algebra system.

Usage

```
sympy(..., retclass = c("character", "Sym", "NULL"), debug = FALSE)
```

Arguments

`...`	Character strings which are pasted together with space separators. The resulting string is passed to sympy.
`retclass`	Character string representing the class of the output or the string "NULL" to mean no output.
`debug`	Logical. If TRUE then additional debugging info is shown.

Details

The `sympy` function passes an input string to SymPy and returns the output. The first time `sympy` is invokved in a session it also starts up SymPy by invoking `sympyStart` (which sets the appropriate paths, calls `jythonStart` and then imports `sympy`). As a result the first invocation of sympy can be expected to much slower than subsequent ones. `jythonStart` creates a variable `.Jython` which is stored in the global environment holding the connection information to the SymPy/Jython session.

Internally if the argument `output=TRUE`, the default, input character string is prefaced with `__Rsympy=` so if such preface would cause an error then ensure that the argument `output=FALSE`.

Note that error messages from SymPy appear on the shell/batch console, not on the R console. In the case of an error message the returned value may be wrong.

Value

The character string produced from SymPy.

Note

SymPy is run under Jython, the Java version of Python.

References

http://code.google.com/p/sympy/, http://www.jython.org/Project/

Examples

```
## Not run:
# These examples are mostly from: http://wiki.sympy.org/wiki/Tutorial

# create a SymPy variable called x
sympy("var('x')")
sympy("y = x*x")
sympy("y")

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

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

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

sympy("xr = Symbol('xr', real=True)")
sympy("exp(I*xr).expand(complex=True)")

# Matrices are stored row by row (unlike R matrices)
cat(sympy("A = Matrix([[1,x], [y,1]])"), "\n")
cat(sympy("A**2"), "\n")
```

```
## End(Not run)
```

Index