

# Résumer pour SYMPY

Jingwen SU

12/20/2020

## Introduction

À la fin du cours, tous les étudiants ont téléchargé les documents pertinents sur Github et partagé les résultats avec nous. Afin de profiter pleinement de ces packages d'installation R sur les mathématiques, je vais lire, analyser et évaluer les articles de mes collègues pour améliorer mes lacunes.

Ceci est une introduction à l'article lu dans cet article. Si vous souhaitez en savoir plus, vous pouvez rechercher des références.

- **Title:** SymPy\_Test\_Alexandre
- **Auteurs:** Alexandre
- **Lien sur Github:** Alexandre

## Synthèse du travail en question

Ceci est un article sur Rsympy. L'auteur présente d'abord le package d'installation, propose d'installer les deux autres logiciels pour l'aider, puis montre comment l'installer. Dans la deuxième partie, l'auteur montre comment utiliser Rsympy pour le calcul. Ce qui comprend principalement: création de variables, limites d'une expression, dérivation de termes, décimales de Pi, simplification or développement or factorisation d'une expression, et résolution d'une équation.

## Contenu principal et explication

### 1. Installation

Parce que SymPy est spécial, il a besoin de deux packages d'installation pour vous aider, nous devons donc installer trois packages d'installation à l'avance lors de son utilisation.

code montrer comme ci-dessous:

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

```
library(rJava)
library(rJython)
```

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

```
library(rSymPy)
```

## 2. Les fonctions

- Création de variables

```
sympy("var('x')") #on introduit une variable x
```

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

```
sympy("y = x*x") #on introduit une variable y équivalente à x au carré
```

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

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

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

- Limites d'une expression

```
sympy("limit(1/x, x, oo)") #limite de 1/X quand x tend vers l'infini (noté; "oo")
```

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

```
sympy("limit(1/x, x, 0)") #limite de 1/X quand x tend vers 0
```

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

- Dérivation de termes

```
sympy("diff(sin(2*x), x, 1)") #dérivation au premier degrés de sin(2x)
```

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

```
sympy("diff(sin(2*x), x, 2)") #dérivation au second degrés de sin(2x)
```

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

- Décimales de Pi

```
sympy("pi.evalf(120)") #nous permet d'afficher les 120 premières décimales de Pi
```

```
## [1] "3.141592653589793238462643383279502884197169399375105820974944592307816406286208998628034825342"
```

- Simplification d'une expression

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

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

- Développement d'une expression

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

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

- Factorisation d'une expression

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

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

- Résolution d'une équation

```
sympy("solve(x**2 - 2, x)") #on résout ici  $x^2-2=0$ 
```

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

## Evaluation et résumer

Selon mes normes, dans l'ensemble, il s'agit d'un article moyen supérieur. Il a un bon format et une introduction générale à rSympy. Il contient des exemples de fonctions et une bibliographie à la fin. Mais la façon dont il explique la fonction est trop directe, bien que nous puissions comprendre de cette façon, je pense que la description dans le texte sera plus belle.