

AstéRix_thomas_duval_rsympy

Teddy Leandre

23/12/2020

I.Présentation

Dans ce dossier, nous chercherons à évaluer 12 travaux comportant du code R et des concepts mathématiques. L'évaluation se fera sur la base de 5 critères de notation qui, additionné donnera une note globale sur 20 points.

1. Qualité du visuel du document sur 4 points
2. Pluralité des fonctionnalités sur 4 points
3. Fonctionnement du code sur 4 points
4. Qualité des explications sur 4 points
5. Lisibilité du code sur 4 points

II. Description

Ici nous verrons l'outil **Rsympy**, c'est un paquet qui permet d'effectuer certains calculs. Le travail de Thomas DUVAL se porte sur certaines des fonctionnalités principales du paquet.
Le Github évalué

a. Commentaire

La librairie **rSymPy** comprend un ensemble de fonctions permettant le traitement et la manipulation d'expressions mathématiques symboliques. Cette librairie nécessite l'installation des librairies **rJython** et **rJava** ainsi que l'installation d'un JDK Java afin de fonctionner pleinement. Nous allons dans ce document observer quelques exemples simples d'utilisation de cette librairie et de ces fonctions.

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

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

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

```
library(rSymPy)
```

```
Sys.setenv(JAVA_HOME='C:/Users/The Moneytizer/jdk-15.0.1') #indiquer chemin du dossier contenant JAVA
```

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)]"
```

Sources:

Ondrej Certik, G Grothendieck (SymPy itself is by, and Contributors: Carlos J. Gil Bellosta others). n. "Package 'rSymPy'." <https://cran.r-project.org/web/packages/rSymPy/rSymPy.pdf>.
<http://www.di.fc.ul.pt/~jpn/r/symbolic/>: "Symbolic Computation in R" par João Neto

Notation

Critère 1 : 3/4 Peut être amélioré.

Critère 2 : 3/4 Utilisation de fonctions essentielles mais reste simple.

Critère 3 : 4/4 tout fonctionne.

Critère 4 : 3/4 Explications brèves sur certaines lignes, voir pas d'explications pour certaines fonctions. Des commentaires supplémentaires peuvent être ajoutés.

Critère 5 : 3/4 très simple et lisible.

III. Conclusion

Note globale 16/20. Les fonctionnalités du paquet présentées sont lisibles et présentables. Les sources sont bien citées