R & Python

Juan David Varela R

5/4/2020

Reticulate

```
library(reticulate)
use_python("/Users/juandavid/anaconda3/bin/python3")
# py_install("Nombre del paquete a instalar")

os <- import("os")
os$listdir(".")
```

```
## [1] "01-EjemploRMD.pdf"    "01-EjemploRMD.Rmd"    "02-Documentacion.pdf"
## [4] "02-Documentacion.Rmd" "add.py"               "Prueba1.log"
## [7] "Prueba1.pdf"          "Prueba1.Rmd"
```

```
source_python("add.py")

add(3, 4)
```

```
## [1] 7
```

```
np <- import("numpy", convert = FALSE)

x <- np$array(c(1:4))
sum <- x$cumsum()

print(sum)
```

```
## [ 1  3  6 10]
```

```
py_to_r(sum)
```

```
## [1]  1  3  6 10
```

Ayuda

```
help(py_to_r)
py_help(os$chdir)
```

Arrays

```r
a <- np_array(c(1:10), order = "C")
a
```

```
## [ 1  2  3  4  5  6  7  8  9 10]
```

```r
datos <- iris
head(datos)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```r
datos_py <- r_to_py(datos)
```

```python
import numpy as np
import pandas as pd

r.datos_py.head()
```

```
##    Sepal.Length  Sepal.Width  Petal.Length  Petal.Width Species
## 0           5.1          3.5           1.4          0.2  setosa
## 1           4.9          3.0           1.4          0.2  setosa
## 2           4.7          3.2           1.3          0.2  setosa
## 3           4.6          3.1           1.5          0.2  setosa
## 4           5.0          3.6           1.4          0.2  setosa
```

Sparse Matrix

```r
library(Matrix)
N <- 6
set.seed(123)
sparse_mat <- sparseMatrix(
  i = sample(N, N, replace = F),
  j = sample(N, N, replace = F),
  x = runif(N),
  dims = c(N, N)
  )
sparse_mat
```

```
## 6 x 6 sparse Matrix of class "dgCMatrix"
##
## [1,] .         .          0.8895393 .         .        .
## [2,] .         0.04205953 .         .         .        .
## [3,] .         .          .         .         0.899825 .
## [4,] .         .          .         .         .        0.3279207
## [5,] 0.9545036 .          .         .         .        .
## [6,] .         .          .         0.2460877 .        .
```

```
sparse_mat_py <- r_to_py(sparse_mat)
```

```
r.sparse_mat_py
```

```
## <6x6 sparse matrix of type '<class 'numpy.float64'>'
##  with 6 stored elements in Compressed Sparse Column format>
```

```
py_to_r(sparse_mat_py)
```

```
## 6 x 6 sparse Matrix of class "dgCMatrix"
##
## [1,] .         .          0.8895393 .         .        .
## [2,] .         0.04205953 .         .         .        .
## [3,] .         .          .         .         0.899825 .
## [4,] .         .          .         .         .        0.3279207
## [5,] 0.9545036 .          .         .         .        .
## [6,] .         .          .         0.2460877 .        .
```

Pregunta 1

```
v1 = c(1, 2, 3, 4, 4, 3, 2, 1, 0, 1, 0, 2, 3, 0, 4, 0)
A = matrix(v1, nrow = 4, byrow = T)

v2 = c(4, 3, 2, 1, 0, 3, 0, 4, 1, 2, 3, 4, 0, 1, 0, 2)
B = matrix(v2, nrow = 4, byrow= T)

# Operaciones

A %*% B
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    7   19   11   29
## [2,]   18   26   14   26
## [3,]    0    5    0    8
## [4,]   16   17   18   19
```

```
B %*% A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   19   19   22   23
## [2,]   24    9   22    3
## [3,]   21   11   23   12
## [4,]   10    3   10    1
```

```
t(A %*% B)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    7   18    0   16
## [2,]   19   26    5   17
## [3,]   11   14    0   18
## [4,]   29   26    8   19
```

```
t(B) %*% A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    4    9   12   18
## [2,]   18   17   19   19
## [3,]    2    7    6   14
## [4,]   23   18   19   16
```

```
solve(A %*% B)
```

```
##       [,1]  [,2]  [,3]  [,4]
## [1,] -1.66 -0.65  4.52  1.52
## [2,]  1.60  0.80 -4.60 -1.60
## [3,]  1.02  0.35 -2.84 -0.84
## [4,] -1.00 -0.50  3.00  1.00
```

```
solve(A) %*% t(B)
```

```
##               [,1] [,2] [,3] [,4]
## [1,]  6.000000e-01  2.4  6.4  1.2
## [2,] -3.330669e-16 -2.0 -7.0 -1.2
## [3,] -2.000000e-01 -0.8 -3.8 -0.4
## [4,]  1.000000e+00  1.0  5.0  0.6
```

$$A \cdot B = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \\ 0 & 1 & 0 & 2 \\ 3 & 0 & 4 & 0 \end{pmatrix} \cdot \begin{pmatrix} 4 & 3 & 2 & 1 \\ 0 & 3 & 0 & 4 \\ 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 2 \end{pmatrix} = \begin{pmatrix} 7 & 19 & 11 & 29 \\ 18 & 26 & 14 & 26 \\ 0 & 5 & 0 & 8 \\ 16 & 17 & 18 & 19 \end{pmatrix}$$

$$B \cdot A = \begin{pmatrix} 4 & 3 & 2 & 1 \\ 0 & 3 & 0 & 4 \\ 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \\ 0 & 1 & 0 & 2 \\ 3 & 0 & 4 & 0 \end{pmatrix} = \begin{pmatrix} 19 & 19 & 22 & 23 \\ 24 & 9 & 22 & 3 \\ 21 & 11 & 23 & 12 \\ 10 & 3 & 10 & 1 \end{pmatrix}$$

Pregunta 2

$$dni = (1, 1, 4, 4, 5, 6, 3, 1, 3, 2)$$

```
dni = c(1, 1, 4, 4, 5, 6, 3, 1, 3, 2)
cuadrado = function(v) {
  v^2
}
raiz = function(v) {
```

```
  sqrt(v)
}

cuadrado(dni)
```

```
##  [1]  1  1 16 16 25 36  9  1  9  4
```

```
round(raiz(dni), 2)
```

```
##  [1] 1.00 1.00 2.00 2.00 2.24 2.45 1.73 1.00 1.73 1.41
```

```
sum(dni)
```

```
## [1] 30
```

$$dni^2 = (1, 1, 16, 16, 25, 36, 9, 1, 9, 4)$$

$$\sqrt{dni} = (1, 1, 2, 2, 2.4, 2.45, 1.73, 1, 1.73, 1.41)$$

$$sum(dni) = 30$$

Pregunta 3

$$NombreCompleto = (J, U, A, N, V, A, R, E, L, A)$$

```
nombre = c("J", "U", "A", "N", "V", "A", "R", "E", "L", "A")
nombre[1:4]
```

```
## [1] "J" "U" "A" "N"
```

```
nombre[5:length(nombre)]
```

```
## [1] "V" "A" "R" "E" "L" "A"
```

```
sort(nombre)
```

```
##  [1] "A" "A" "A" "E" "J" "L" "N" "R" "U" "V"
```

```
rbind(nombre[1:4], nombre[5:length(nombre)])
```

```
## Warning in rbind(nombre[1:4], nombre[5:length(nombre)]): number of columns
## of result is not a multiple of vector length (arg 1)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] "J"  "U"  "A"  "N"  "J"  "U"
## [2,] "V"  "A"  "R"  "E"  "L"  "A"
```

$$Nombre = (J, U, A, N)$$
$$Apellido = (V, A, R, E, L, A)$$

$$OrdenAlfabético = (A, A, A, E, J, L, N, R, U, V)$$

$$MatrizNombres = \begin{pmatrix} J & U & A & N & J & U \\ V & A & R & E & L & A \end{pmatrix}$$