

Chunk y Variables

Tony

R Markdown

Cargar la librería **reticulate** para usar *Phyton* en este RMD. En particular, se ocupará la carpeta de librerías de **Anaconda** y no las que trae R. También veremos los archivos en el directorio de trabajo usando **Phyton**. En lugar del punto `.` para usar los métodos se usa `$`.

```
library(reticulate)
use_python("/home/usuario/anaconda3/bin/python3")
# Si queremos instalar un paquetes de python:
# py_install("nombre del paquete a instalar")
os <- import("os")
```

```
## Warning: Python '/home/usuario/anaconda3/bin/python3' was requested but '/
## home/usuario/anaconda3/envs/r-reticulate/bin/python' was loaded instead (see
## reticulate::py_config() for more information)
```

```
os$listdir(".")
```

```
## [1] "Matrices.Rmd"          "Variables_ocultas.html" "Espacios_en_blanco.Rmd"
## [4] "Espacios_en_blanco.pdf" "Variables_ocultas.Rmd"  "Espacios_en_blanco.tex"
## [7] "01-EjemploRMD.Rmd"     "Variables_ocultas.pdf"  "02-Documentacion.Rmd"
## [10] "02-Documentacion.html" "R_and_Phyton.Rmd"       "02-Documentacion.pdf"
## [13] "R_and_Phyton.pdf"       "test.tex"               "Chunks.Rmd"
## [16] "Chunks.pdf"             "R_and_Phyton.html"       "Auxiliares.pdf"
## [19] "01-EjemploRMD.html"     "test.Rmd"                "add.py"
## [22] "Auxiliares.Rmd"         "R_and_Phyton.md"         "Matrices.pdf"
## [25] "01-EjemploRMD.pdf"      "test.pdf"
```

Cargar un módulo o importar una función

Para cargar un módulo o función se escribe: `\textbf{source_python("nombre del modulo")}`.

```
source_python("add.py")
add(3,4)
```

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

Importar paquetes

El código es: `import "numpy", convert=FALSE` Con el parámetro `convert=FALSE` indicamos que no se hará ninguna transformación de objetos de *Phyton* a objetos de R, sino que solo se trabajarán con objetos nativos de *Phyton*. Con la función `\textbf{py_to_r()}` convertimos objetos de *Phyton* a objetos de R.

```
# Importamos el paquete con un apodo
np <- import("numpy", convert=FALSE)
x <- np$array(c(1:4))
sum <- x$cumsum()
# Se aprecia que sum es una variable de Python ya que aparece el array entre corchetes
print(sum)
```

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

```
# Haciendo la conversión a R después de operar en Python
py_to_r(sum)
```

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

Función de ayuda

Sombrear la sección de código y ejecutar individualmente.

```
# R:
help(py_to_r)
# Python
py_help(os$chdir)
```

Arrays

Con **order="C"** se indica que el objeto se guarde en memoria con el estilo de C en lugar del de Fortran.

```
a <- np_array(c(1:10), dtype="float16")
a
```

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

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

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

Dataframes

Convertiremos un dataframe de R a un dataframe de Python:

```
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
```

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

```
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
```

Matriz

Convertiremos un dataframe de R a un dataframe de Python para luego convertirla a un objeto de R:

```
library(Matrix)
mat <- matrix(1:12, nrow = 3, ncol = 4)
mat
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    4    7   10
## [2,]    2    5    8   11
## [3,]    3    6    9   12
```

```
mat_py <- r_to_py(mat)
```

```
r.mat_py
```

```
## array([[ 1,  4,  7, 10],
##        [ 2,  5,  8, 11],
##        [ 3,  6,  9, 12]], dtype=int32)
```

```
py_to_r(mat_py)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    4    7   10
## [2,]    2    5    8   11
## [3,]    3    6    9   12
```