R & Python

Kiko

1/7/2021

Reticulate

```
library(reticulate)
use_python("C:/ProgramData/Anaconda3/envs/CursoEstadisticaDescriptiva/python.exe")

os <- import("os")
```

```
## Warning: Python 'C:/ProgramData/Anaconda3/envs/CursoEstadisticaDescriptiva/
## python.exe' was requested but 'C:/ProgramData/Anaconda3/python.exe' was loaded
## instead (see reticulate::py_config() for more information)
```

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

matrix(x, length(x))
```

```
##       [,1]
##  [1,] "01-EjemploRMD.pdf"
##  [2,] "01-EjemploRMD.Rmd"
##  [3,] "02-Documentación.pdf"
##  [4,] "02-Documentación.Rmd"
##  [5,] "kiko.png"
##  [6,] "RyPython.html"
##  [7,] "RyPython.pdf"
##  [8,] "RyPython.Rmd"
##  [9,] "suma.py"
```

```
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

El resultado de i^2:

```
from sympy import *
for i in range(10):
  print("$"+latex(str(pow(i,2)))+"$")
```

0 1 4 9 16 25 36 49 64 81

Arrays

```
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
```

```
library(Matrix)
N <- 6
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,] .         0.6405617 .        .         .         .
## [2,] .         .         0.457131 .         .         .
## [3,] .         .         .        0.6047847 .         .
## [4,] .         .         .        .         0.6465559 .
## [5,] 0.3283821 .         .        .         .         .
## [6,] .         .         .        .         .         0.07959798
```

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

```
print(r.sparse_mat_py)
```

```
##   (4, 0) 0.3283820804208517
##   (0, 1) 0.6405617324635386
##   (1, 2) 0.4571310121100396
##   (2, 3) 0.604784700088203
##   (3, 4) 0.6465558602940291
##   (5, 5) 0.07959798141382635
```