

Python para Ciência de Dados

Reticulate



6 de 2022

Motivação

- Já aprendemos muito sobre Python e sobre como configurá-lo
 - Google Colab, notebooks, etc.
 - Instalação local, Conda, Virtualenv, etc.
 - PyCharm, VS Code, etc.
- Mas não queremos jogar fora o nosso conhecimento de R!
 - Não precisa de Conda/Virtualenv e funciona no RStudio
 - Será possível unir o útil ao agradável?
- Queremos:
 - Aproveitar código Python que já existe
 - Continuar usando R onde for mais confortável
 - Não precisar gerenciar ambientes ou trocar de IDE

Introdução

Reticulate

- Reticulate é um pacote com todas as ferramentas necessárias para conectar o R com o Python
 - Chamar o Python pelo R (scripts, módulos, etc.)
 - Traduzir objetos entre o R e o Python
 - Conexão direta com ambientes virtuais (Conda, Virtualenv)
- Muito útil para interoperabilidade entre as duas linguagens
 - Times que usam R e Python
 - Uso de funcionalidades exclusivas à outra linguagem
 - Centralização de workflows
- Basta instalar!

```
install.packages("reticulate")
```

Uso básico

```
library(reticulate)
```

```
# Inicializar o Python do sistema  
py_available(initialize = TRUE)
```

```
## [1] TRUE
```

```
# Verificar versão  
py_version()
```

```
## [1] '3.8'
```

```
# Executar comando Python e pegar resultado no R  
py_eval("[1, 2, 3] + [4]")
```

```
## [1] 1 2 3 4
```

Conda e Virtualenv

- O reticulate é capaz de instalar o Miniconda

```
install_miniconda()
```

```
# Versão do Conda instalado pelo Miniconda  
conda_version()
```

```
## [1] "conda 4.13.0"
```

- Ele não consegue instalar o Virtualenv, mas consegue usá-lo (só no Mac e no Linux!)

```
# Raíz onde os ambientes do Virtualenv vão ficar  
virtualenv_root()
```

```
## [1] "~/virtualenvs"
```

Usando o ambiente virtual

- Camos fazer tudo no Conda em no ambiente `r-reticulate`
 - A recomendação é não desviar do padrão

```
# Feito uma vez só (= conda create)  
conda_create("r-reticulate")
```

```
# Rodar no começo de cada sessão (= conda activate)  
use_condaenv("r-reticulate")
```

```
# Instalar pacote no r-reticulate (= conda install)  
py_install("platform")
```

```
# Verificar se o pacote foi instalado  
py_module_available("platform")
```

```
## [1] TRUE
```

Uso

Chamando o Python

- Há 4 jeitos de chamar o Python depois de feita a configuração
 - Executar um arquivo inteiro com `source_python()` da mesma forma como faríamos com `source()`
 - Importação direta de módulos dentro do R
 - Dentro do RMarkdown com comunicação bidirecional
 - REPL (Read, Eval, Print Loop) no Console do RStudio
- Todos esses métodos usam o ambiente `r-reticulate` do Conda como padrão
 - No caso de qualquer problema, a função `py_discover_config()` é sua amiga

Import

- A importação ocorre como no Python: um objeto com funções
 - Ao invés do `.`, usamos `$`
 - A sintaxe vira uma mistura de R e Python (vide exemplo)
 - É possível criar wrappers de pacotes do Python!

```
# Pegar nome da plataforma  
platform <- import("platform")  
platform$platform()
```

```
## [1] "macOS-12.4-arm64-arm-64bit"
```

```
# Pegar primeiro elemento de uma array do Numpy  
np <- import("numpy")  
np$array(c(1, 2, 3, 4))[0]
```

```
## numeric(0)
```

Conversões

- Código em R e o que ele vira em Python
 - Atenção especial aos índices! No Python eles começam em 0

| Exemplo | Python |
|--|-------------------|
| <code>1, 1L, TRUE, "foo"</code> | Escalar |
| <code>c(1.0, 2.0, 3.0), c(1L, 2L, 3L)</code> | Lista |
| <code>list(1L, TRUE, "foo")</code> | Tupla |
| <code>list(a = 1L, b = 2.0), dict(x = x_data)</code> | Dicionário |
| <code>matrix(c(1,2,3,4), nrow = 2, ncol = 2)</code> | NumPy ndarray |
| <code>data.frame(x = c(1,2,3))</code> | Pandas DataFrame |
| <code>function(x) x + 1</code> | Função |
| <code>as.raw(c(1:10))</code> | Python bytearray |
| <code>NULL, TRUE, FALSE</code> | None, True, False |

RMarkdown

- Parece um truque de magia

```
# Código Python normal  
obj = [1, 2, 3]  
print(obj)
```

```
## [1, 2, 3]
```

```
# Código R normal  
obj2 <- py$obj + 1  
print(obj2)
```

```
## [1] 2 3 4
```

```
# Código Python normal  
r.obj2 + [5.0]
```

```
## [2.0, 3.0, 4.0, 5.0]
```

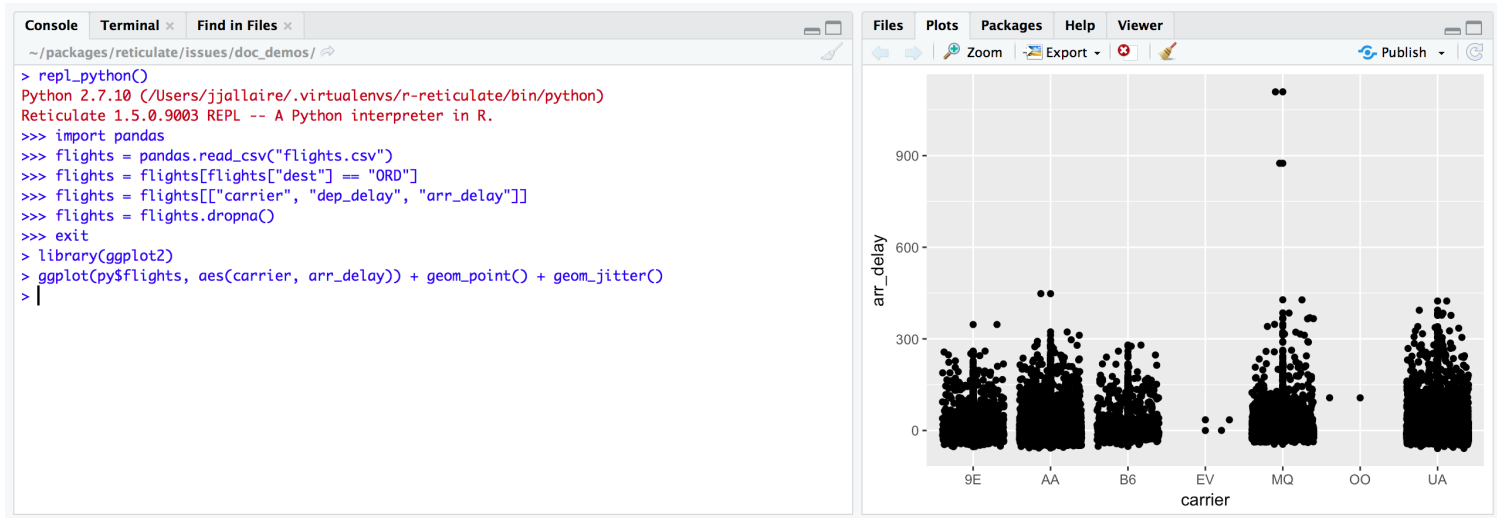
RMarkdown

- Basta especificar a linguagem no começo do chunk
 - Os objetos do Python ficam acessíveis em `py$x`
 - Os objetos do R ficam acessíveis em `r.x`

```
13
14 ▾ ```{python}
15 import pandas
16 flights = pandas.read_csv("flights.csv")
17 flights = flights[flights['dest'] == "ORD"]
18 flights = flights[['carrier', 'dep_delay', 'arr_delay']]
19 flights = flights.dropna()
20 ```
21
22 ▾ ```{r, fig.width=7, fig.height=3}
23 library(ggplot2)
24 ggplot(py$flights, aes(carrier, arr_delay)) + geom_point() + geom_jitter()
25 ```
26
```

REPL

- REPL (Read, Eval, Print Loop) é o termo técnico para ferramentas como o "Console" que usamos no R
 - Se abrirmos um arquivo Python no RStudio e rodarmos qualquer linha com CTRL + ENTER, vamos ativar o REPL do Python
 - Ao invés do ">" tradicional, o prompt vira ">>>"
 - Para sair basta rodar o comando `exit` e voltaremos para o R



REPL

Syntax highlighting for Python scripts and chunks.

Tab completion for Python functions and objects (and Python modules imported in R scripts).

Source Python scripts.

Execute Python code line by line with **Cmd + Enter** (**Ctrl + Enter**).

View Python objects in the Environment Pane.

View Python objects in the Data Viewer.

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains Python code for importing libraries and plotting data. A tab completion menu is visible for the `tips` object.
- Environment Pane:** Shows the current environment with objects like `r`, `tips`, `mpl`, `pd`, `sns`, and `sys`.
- Data Viewer:** Displays a table of data from the `tips` DataFrame.
- Console:** Shows the output of the `reticulate::repl_python()` command, indicating that a Python REPL is running.
- Plots Pane:** Displays a scatter plot of `total_bill` vs `tip` with a regression line, colored by `smoker` status.

| | total_bill | tip | sex | smoker | day | time | size |
|---|------------|------|--------|--------|-----|--------|------|
| 1 | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 |
| 2 | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 |
| 3 | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 |

A Python REPL opens in the console when you run Python code with a keyboard shortcut. Type **exit** to close.

matplotlib plots display in plots pane.

Press **F1** over a Python symbol to display the help topic for that symbol.

Materiais extras

- Colinha do reticulate
- Documentação completa do reticulate