

CT5102: Programming for Data Analytics

Lecture 12: Calling Python from R using reticulate

Dr. Jim Duggan,
School of Engineering & Informatics
National University of Ireland Galway.
<https://github.com/JimDuggan/CT5102>



Overview

- Overall approach
- Creating a package in Python
- Loading into R
- Running the Python functions & benchmark

<https://rstudio.github.io/reticulate/>

reticulate

Home

Articles ▾

Reference

News

R Interface to Python

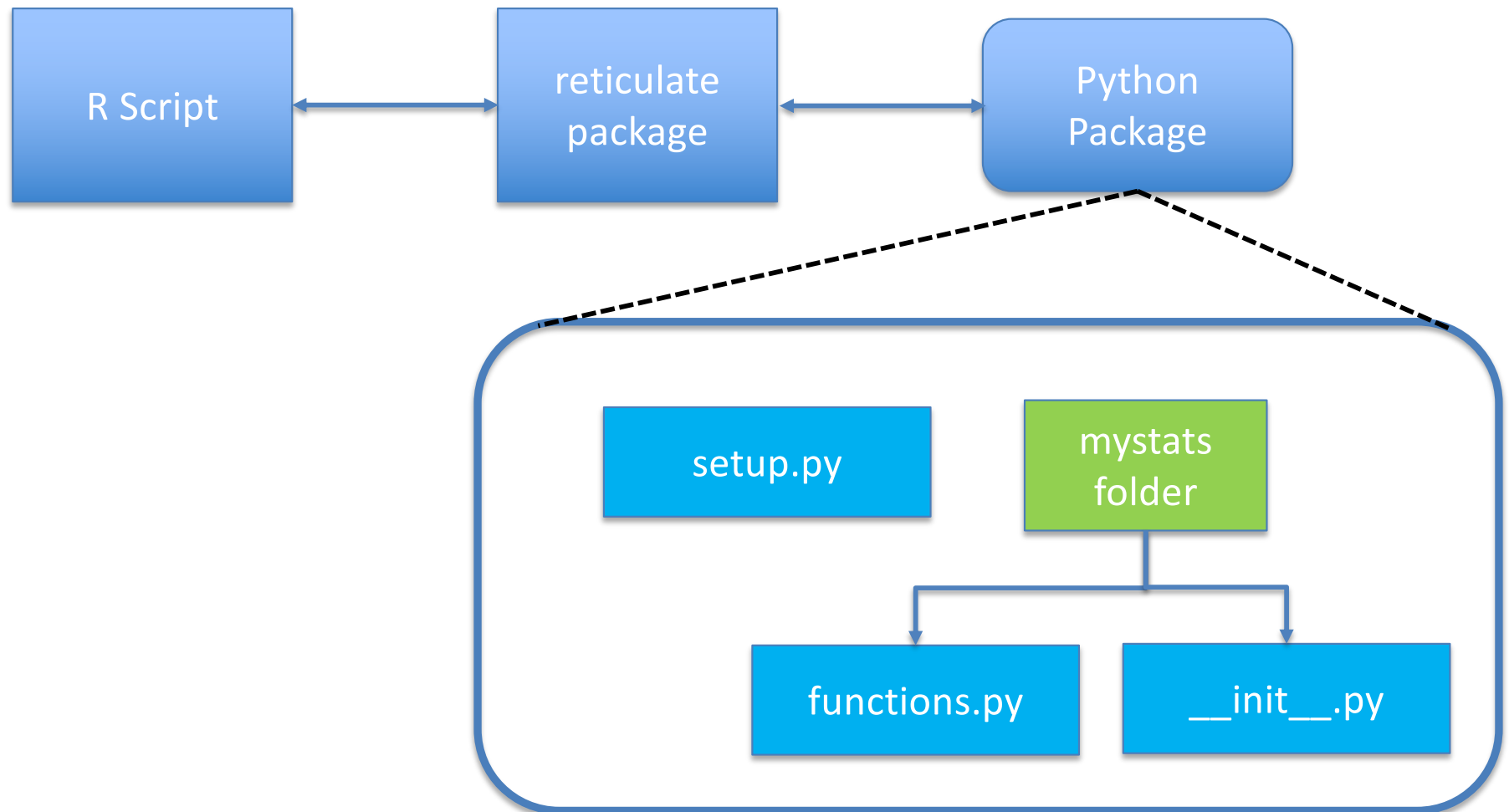
The **reticulate** package provides a comprehensive set of tools for interoperability between Python and R. The package includes facilities for:

- Calling Python from R in a variety of ways including R Markdown, sourcing Python scripts, importing Python modules, and using Python interactively within an R session.
- Translation between R and Python objects (for example, between R and Pandas data frames, or between R matrices and NumPy arrays).
- Flexible binding to different versions of Python including virtual environments and Conda environments.

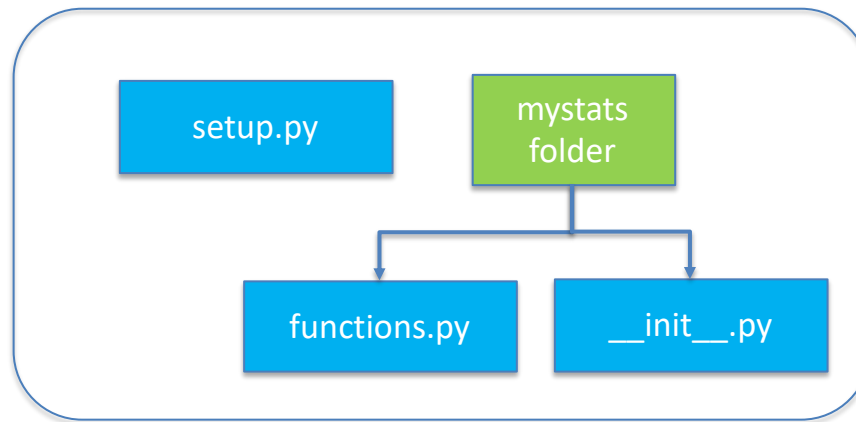


Reticulate embeds a Python session within your R session, enabling seamless, high-performance interoperability. If you are an R developer that uses Python for some of your work or a member of data science team that uses both languages, reticulate can dramatically streamline your workflow!

Overall Approach



Sample Python Package mystats



```
× __init__.py × functions.py × setup.py
1 # To create the package:
2 #   python3 setup.py install
3
4 from setuptools import setup
5 setup(name='mystats',
6       version=0.1,
7       description='Test package for R',
8       url='#',
9       author='JD',
10      license='MIT',
11      packages=['mystats'],
12      zip_safe=False)
13
14
```

```
× __init__.py × functions.py × setup.py*
1 from .functions import summary
2
3
```

```
× __init__.py × functions.py × setup.py*
1 import statistics
2
3 def summary(v):
4     ans = {'Mean' : statistics.mean(v),
5           'SD' : statistics.stdev(v),
6           'Median' : statistics.median(v)}
7     return ans
8
9
```

Building the package for system-wide use

```
[(base) MyMacBook:python jim$ python3 setup.py install
running install
running bdist_egg
running egg_info
creating mystats.egg-info
writing mystats.egg-info/PKG-INFO
writing dependency_links to mystats.egg-info/dependency_links.txt
```

```
Python 3.7.3 (default, Mar 27 2019, 16:54:48)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from mystats import summary
>>> summary((1,2,3))
{'Mean': 2, 'SD': 1.0, 'Median': 2}
```

R Code

```
1 library(reticulate)
2 Sys.setenv(RETICULATE_PYTHON="/Users/jim/anaconda3/bin/python3")
3
4
5 my_summary <- function(v){
6   list(Mean=mean(v),
7        SD=sd(v),
8        Median=median(v))
9 }
10
11 reticulate::py_config()
12
13 mystats <- reticulate::import("mystats")
14
15 obs <- rnorm(1000,78,12)
16
17 from_python <- mystats$summary(obs)
18 with_R <- my_summary(obs)
```

Comparing Output

```
> from_python
```

```
$Mean
```

```
[1] 77.75083
```

```
$SD
```

```
[1] 11.78844
```

```
$Median
```

```
[1] 77.68424
```

```
> with_R
```

```
$Mean
```

```
[1] 77.75083
```

```
$SD
```

```
[1] 11.78844
```

```
$Median
```

```
[1] 77.68424
```


Type Conversions

R	Python	Examples
Single-element vector	Scalar	1, 1L, TRUE
Multi-element vector	List	c(1.0,2.0,3.0)
List of multiple types	Tuple	list(1L,TRUE,"foo")
Named list	Dict	List(a=1,b=2)
Matrix/array	NumPy Array	matrix(1:4,nrow=2)
Data Frame	Pandas Data Frame	data.frame()

- When calling into Python, R data types are automatically converted to their equivalent Python types
- When values are returned from Python, they are converted back to R types

Further information

<https://blog.rstudio.com/2018/03/26/reticulate-r-interface-to-python/>

Built in conversion for many Python object types is provided, including [NumPy](#) arrays and [Pandas](#) data frames. From example, you can use Pandas to read and manipulate data then easily plot the Pandas data frame using [ggplot2](#):

```
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
```

Summary

- Facilitates function calls from R to Python
- Many objects automatically translated
- Provide mechanism for also integrating Python with RShiny



<https://github.com/JimDuggan/CT5102/tree/master/code/course/12%20reticulate>