R and Python in one Jupyter notebook

34

24

Is it possible to run R and Python code in the same Jupyter notebook. What are all the alternatives available?

1. Install r-essentials and create R notebooks in Jupyter.
2. Install rpy2 and use rmagic functions.
3. Use a beaker notebook.

Which of above 3 options is reliable to run Python and R code snippets (sharing variables and visualizations) or is there a better option already?

python r python-2.7 ipython jupyter-notebook

asked Aug 18 '16 at 0:09

Sailendra Pinupolu
381 1 3 8

5 Answers

36

Yes, it is possible! Use rpy2.

You can install rpy2 with: `pip install rpy2`

Then run `%load_ext rpy2.ipython` in one of your cells. (You only have to run this once.)

Now you can do the following:

Python cell:

```
# enables the %%R magic, not necessary if you've already done this
%load_ext rpy2.ipython

import pandas as pd
df = pd.DataFrame({
    'cups_of_coffee': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
    'productivity': [2, 5, 6, 8, 9, 8, 0, 1, 0, -1]
})
```

R cell:

```
%%R -i df -w 5 -h 5 --units in -r 200
# import df from global environment
# make default figure size 5 by 5 inches with 200 dpi resolution

install.packages("ggplot2", repos='http://cran.us.r-project.org', quiet=TRUE)
library(ggplot2)
ggplot(df, aes(x=cups_of_coffee, y=productivity)) + geom_line()
```

And you'll get your pretty figure plotting data from a python Pandas DataFrame.

edited Jun 12 '18 at 3:41

answered Jan 27 '17 at 8:50

uut
700 5 14

11

Using @uut's answer for running R in a jupyter notebook within python kernel (in MacOS), the following worked for me.

%%R should always be at the start of the cell else you will get the error as shown in figure below

In [1]:
```
import pandas as pd
df = pd.DataFrame({
    'cups_of_coffee': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
    'productivity': [2, 5, 6, 8, 9, 8, 0, 1, 0, -1]
})
```

In [3]:
```
# enables the %%R magic, not necessary if you've already done this
%load_ext rpy2.ipython
```

In [7]:
```
# import df from global env
# make default figure size 5 by 5 inches, you can change the units to px, cm, etc
%%R -i df -w 5 -h 5 --units in -r 200
install.packages("ggplot2", repos='http://cran.us.r-project.org', quiet=TRUE)
library(ggplot2)
ggplot(df, aes(x=cups_of_coffee, y=productivity)) + geom_line()
```

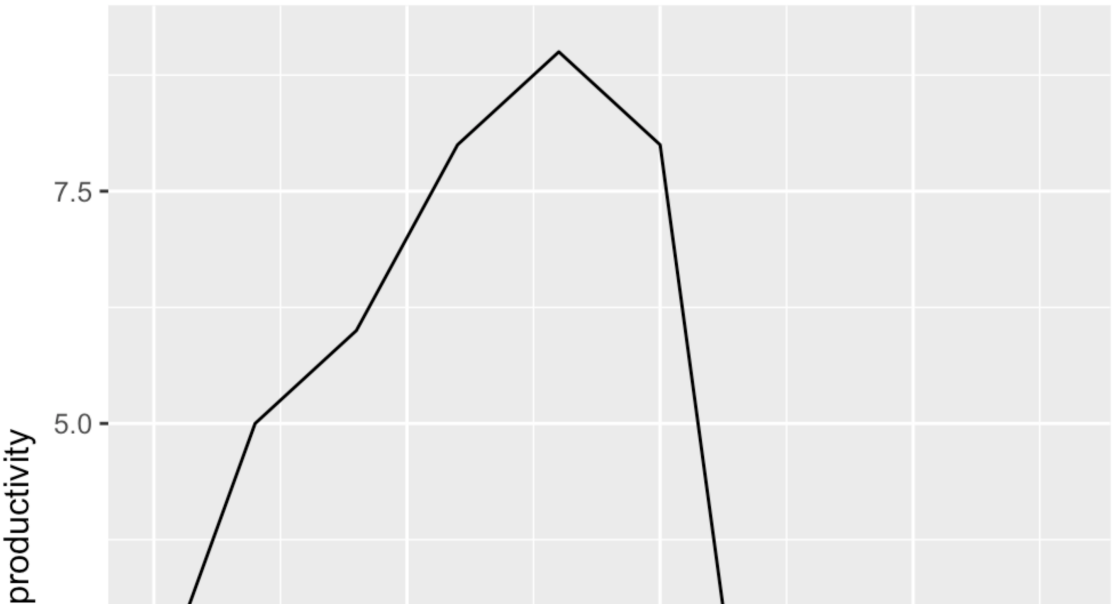
```
  File "<ipython-input-7-480c352382e8>", line 3
    %%R -i df -w 5 -h 5 --units in -r 200
    ^
SyntaxError: invalid syntax
```

The following is the right way:

In [1]:
```
import pandas as pd
df = pd.DataFrame({
    'cups_of_coffee': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
    'productivity': [2, 5, 6, 8, 9, 8, 0, 1, 0, -1]
})
```

In [3]:
```
# enables the %%R magic, not necessary if you've already done this
%load_ext rpy2.ipython
```

In [8]:
```
%%R -i df -w 5 -h 5 --units in -r 200
install.packages("ggplot2", repos='http://cran.us.r-project.org', quiet=TRUE)
library(ggplot2)
ggplot(df, aes(x=cups_of_coffee, y=productivity)) + geom_line()
```



Also `%load_ext rpy2.ipython` should come before `%%R` hence put it in a different cell above it as shown in the figures.

edited Jan 17 '18 at 16:58

answered Jan 17 '18 at 0:06

Abhimanu Kumar

962 10 18

4

UPDATE April 2018,

RStudio has also put out a package: https://blog.rstudio.com/2018/03/26/reticulate-r-interface-to-python/

for which it is possible to run multiple code chunks in different languages using the R markdown notebook, which is similar to a jupyter notebook.

In my previous post, I said that the underlying representation of objects is different. Actually here is a more nuanced discussion of the underlying matrix representation of R and python from the same package: https://rstudio.github.io/reticulate/articles/arrays.html

Old post:

It will be hard for you to use both R and Python syntax in the same notebook, mostly because the underlying representation of objects in the two languages are different. That said, there is a project that does try to allow conversion of objects and different languages in the same notebook: http://beakernotebook.com/features

I haven't used it myself but it looks promising

edited Apr 29 '18 at 14:47

answered Aug 18 '16 at 0:15

Allen Wang
981 9 24

2

SoS kernel is another option.

Don't know how well it performs yet, just started using it.

The SoS kernel allows you to run different languages within the same notebook, including Python and R.

SoS Polyglot Notebook - Instructions for Installing Desired Languages

Here is an example of a notebook with Python and R cells.

*Update:

In terms of sharing variables, one can use the magics `%use` and `%with`. "SoS automatically shares variables with names starting with **sos** among all subkernels"1.

Ex.

Starting cell in R:

```
%use R
sos_var=read.csv('G:\\Somefile.csv')
dim(sos_var)
```

Output:

```
51  13
```

Switching to python:

```
%with Python3
sos_var.shape
```

Output:

```
(51, 13)
```

edited Aug 23 '18 at 15:01

answered Aug 22 '18 at 17:18

rm1104
41 6

---

sos looks nice, but currently the practical use for a developer looks limited. E.g. if you do `%get jl --from`

`julia-1.0 print( jl )` -> Subkernel julia-1.0 does not support magic %put. If you use conda it's had to combine Python and R in one environment. – InLaw Sep 2 '18 at 11:13

-3

I would not recommend to use two languages in a single Notebook. Instead, you can orchestrate R and Python code in project level by connecting them on input\output file base. Data science tools like DVC can help you in order to do that.

You might find some code examples in this blog post: Best practices of orchestrating Python and R code in ML projects

answered Oct 8 '17 at 7:42

Dmitry Petrov
552 7 22

7 This is as non-answer and you are just supporting projects you work on. see the git repo you link. At the very least explain why your "instead" statement is worth following. – mnky9800n Feb 15 '18 at 17:32