Mix R and Python Up

GYH

In order to ensure you to avoid errors as much as possible, I suggest you choose a RStudio version at least 1.2+. Unfortunately, the latest available version on the conda cloud is 1.1+, which means you will meet some trouble with the following operations. But still we have some options: you can see here to fetch the new version for RStudio matching your own operating system. I personally use 1.4.52 on Linux.

To begin with, you should set your python work place. If you use Anaconda, you can set it in your virtual environment as the following way (which means you are able to use python packages by linking your python engine to the customed environment):

```
#```{r setup, include=TRUE}
knitr::opts_chunk$set(echo = TRUE)
library(reticulate)
use_python("/home/gyh/anaconda3/envs/PR/bin/python3")
py_available(initialize=TRUE)
```

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

```
#```
```

Then, the following code shows how python chunk calls the r variables. You can use `r.var` to do that.

```
#```{r}
aaa = 1
#```
```

```
#```{python}
print(r.aaa)
#```
```

```
## 1.0
```

Next, we will see how the r chunk calls the python variables. You can use py$var in r environment to get the variable which is addressed or defined in python.

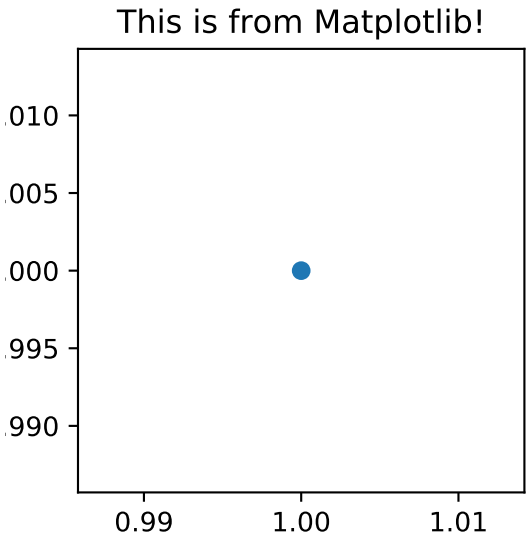
```
#```{python}
bbb = 1
#```
```

```
#```{r}
print(py$bbb)
```

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

Finally, also most attractively, you can use any python package which has been installed in advance in your conda virtual environment. For instance, the following example shows how to call the `Matplotlib` in python chunk.

```
#```{python,fig.height = 3, fig.width = 3,fig.align = "center"}
import matplotlib.pyplot as plt
plt.scatter([1],[1])
plt.title('This is from Matplotlib!')
plt.show()
#```
```



Now, above-mentionedly, we have summarized some useful way to mix python and r up in RStudio with `reticulate` package. Besides, it's also possible to calling r in jupyter notebook with rpy2. We will discuss on that later.

Hope this guide can help you!