Prepare a python string for R using rpy2

Asked 4 years, 3 months ago Active 4 years, 3 months ago Viewed 707 times



This question relates to <u>python variable to R</u> and perhaps also to this <u>python objects to rpy2</u> but none of the two completely overlaps and the first one is actually unanswered.

0

My question is actually very simple. I have a string, say:



```
In [21]: strg
Out[21]: 'I want to go home'
```

and I want to pass it to R through robjects.r(''' ''') like this, for example:

```
robjects.r('''
test <- gsub("to", "",strg)
''')</pre>
```

but of course, when I run this I obtain: Error in gsub("me", "", strg): object 'strg' not found .

I have not used rpy2 much (as obvious) but I guess is related to the environments in which R and Python see the objects.

I have tried a few things, like transforming the string strg to an robject first and then feed it to robjects.r(''') but I get the same error message. Overall, I do not know how do this so that strg is seen at the R environment.

Any help is much appreciated!

Thanks in advance for your time!

```
python r string rpy2
```



asked Jun 10 '15 at 11:25

Javier

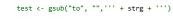
730 1 13 38

2 Answers



Just add the strg value to the command string:







or, by using .format :

robjects.r('''

```
robjects.r('''
test <- gsub("to", "",%s)
'''.format(strg))</pre>
```

Do note that you'll need to watch out for backslashes, see the question here



answered Jun 10 '15 at 11:33

433 2 5

Hi there hajtos. the two of them throw a parsing error: ValueError: Error while parsing the string. The first at the last line ----> 5 ''') coming from --> 310 p = rinterface.parse(string) and the second also at the last line: ----> 5 '''.format(strg)) coming again from the same function --> 310 p = rinterface.parse(string) at def __call__(self, string): - Javier Jun 10 '15 at 11:45 \$\rho\$'

Hi hajtos, I found the reason of the error, which is the *quotes needed for the R expression to work*. so I simply have to define strg = '"I want to go home" and it worked. THANKS! — Javier Jun 10 '15 at 11:58 /



I'd recommend you to create an function, as the R function exposed by rpy2 can be called just as if it was a Python function.



```
my_func = robjects.r('''
function(strg) {
    test <- gsub("to", "",strg)
    test
}
''')
my_func(strg)</pre>
```

answered Jun 12 '15 at 21:52

| Igautier | 9,284 | 21 | 39