Python + LyX

github.com/kenjisato/pythonlyx

May 16, 2020

Set an appropriate Python path in _knitr/setup.R. You may also have to install several R packages, in particular, knitr and reticulate, before you can successfully compile this LyX file. Since using LyX makes debugging very hard, it is advisable for you **not** to use this package if you have little experience in R and LaTeX and if you have a close deadline.

```
import sys
print(sys.version)
```

```
3.7.7 (default, Mar 26 2020, 10:35:24)
[Clang 4.0.1 (tags/RELEASE_401/final)]
```

To include code from an external file, use R function, knitr::read_chunk. Since I set the default engine to python in _knitr/setup.R, you need to explicity specify engine=’R’. You may usually want to make this code invisible in the final PDF file by setting include=FALSE.

```
# Read Python code from within the R session
knitr::read_chunk('include.py')
```

Here’s the content of include.py. As you can see, a section starts with a special line #---- label and ends with a beginning of next section or the end of the file.

```
cat include.py
```

```
#---- code1
import pandas as pd
frame = pd.DataFrame({'x': [1, 2, 3],
                      'y': [0.1, 0.2, 0.3]})
frame

#---- code2
import matplotlib.pyplot as plt
frame.plot()
plt.show()

#---- code3
r = 10

#---- code4
r * 5
```

You can run section code1 by specifying the label.

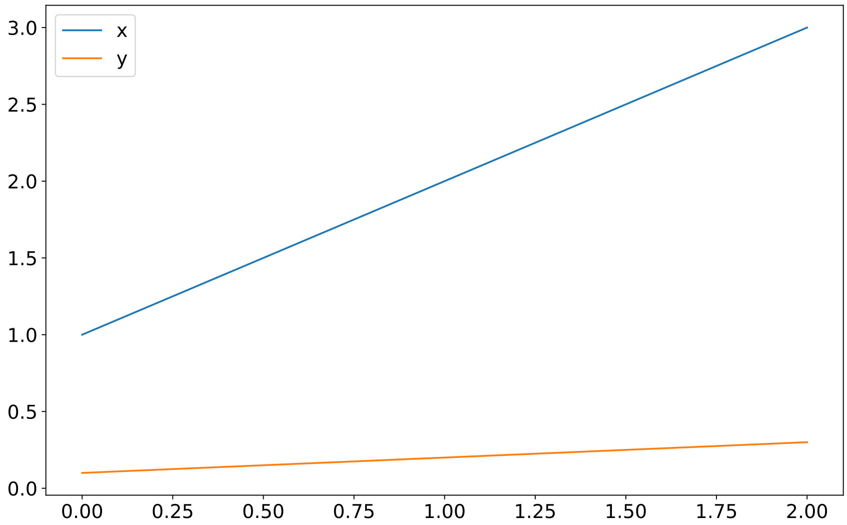

Figure 1: Simple plot

```
import pandas as pd
frame = pd.DataFrame({'x': [1, 2, 3],
                      'y': [0.1, 0.2, 0.3]})
frame
```

```
   x    y
0  1  0.1
1  2  0.2
2  3  0.3
```

For a code that generates a graphic, I prefer to interfere with the default behavior of knitr printing the figure by setting `fig.show='hide'`. I then manually include the figure with `\includegraphics`.

```
import matplotlib.pyplot as plt
frame.plot()
plt.show()
```

You may also wonder whether you can use `r` as a variable name. Since `reticulate` uses `r` to communicate with R objects from within Python code, there is normally a rectriction that you cannot share an object named `r` in multiple chunks. My `_knitr/setup.R` defines a shim for `reticulate::py_inject_r()` to change the name of the special object because I want to use `r` for other purposes.

```
r = 10
```

You can reuse this normally.

```
r * 5
```

```
50
```

You can also use Python variables inline through the special R object, `py`, which passes Python variables to the R session. On the LyX menu, navigate through Insert > Custom insets > S/R expression, then put `py$r` in the box. You have: 10.