Python code chunks in R Markdown: a minimal example

Oliver Sheridan-Methven

15th May 2019

A normal R code chunk

```
library(reticulate)
x = 42
print(x)
```

```
## [1] 42
```

Modify an R variable

In the following chunk, the value of `x` on the right hand side is 42, which was defined in the previous chunk.

```
x = x + 12
print(x)
```

```
## [1] 54
```

A Python chunk

This works fine and as expected.

```
x = 42 * 2
print(x)
```

```
## 84
```

The value of `x` in the Python session is 84. It is not the same `x` as the one in R.

Modify a Python variable

```
x = x + 18
print(x)
```

```
## 102
```

Retrieve the value of `x` from the Python session again:

```
py$x
```

```
## [1] 102
```

Assign to a variable in the Python session from R:

```
py$y = 1:5
```

See the value of `y` in the Python session:

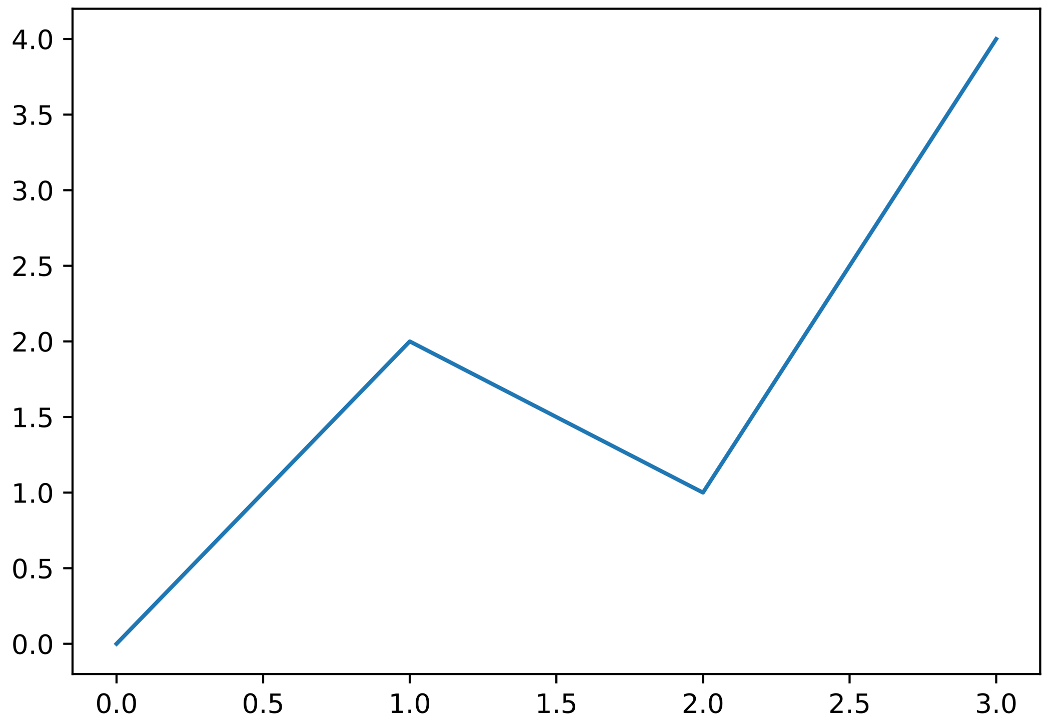
```
print(y)
```

```
## [1, 2, 3, 4, 5]
```

Python graphics

You can draw plots using the **matplotlib** package in Python.

```
import matplotlib.pyplot as plt
plt.plot([0, 2, 1, 4])
plt.show()
```


4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0