

Replace Pweave using Rmarkdown

John Minter

2019-05-09

Abstract

Rmarkdown has replaced Sweave for most documents. This document was composed to mimic the test document developed by John Cook to highlight reproducible documents with python using Sweave. This Rmarkdown document implements code cnks in both python and R to generate a PDF. One needs to set the path to the python executable in the first chunk. Inline python does not work, but we can use inline R to display python results.

Contents

1 Introduction	1
2 Note the R setup chunk	1
3 Import the python packages and functions	2
4 Implement John Cook's examples	2
5 Grapics are important	2

1 Introduction

In 2012 [John Cook](#) explained how to make reproducible reports in Python using Pweave to process the code chunks. The entire document is composed in \LaTeX .

Rmarkdown in RStudio integrates an environment where most text is written in markdown. \LaTeX is supported and is used mostly for mathematics. One can also use the same document to generate documents in PDF, HTML, and Word formats. The conversion is handled by pandoc. This document implements John Cook's exemplar in Rmarkdown.

2 Note the R setup chunk

```
knitr::opts_chunk$set(echo = TRUE, comment=NA)
library(reticulate)
use_python("/Users/jrminter/miniconda3/bin/python")
```

The **r setup** chunk is used to configure R. It is frequently hidden. The reticulate does the heavy lifting for us, so we import the package in the **r setup** code chunk and supply the path to our python distribution. If we needed other R packages, we would load them there.

3 Import the python packages and functions

```
import matplotlib.pyplot as plt
from numpy import pi, linspace, sqrt, sin
```

4 Implement John Cook's examples

Code that sets the value of the variable a .

```
a = 3.14
```

Visible code that sets b and squares it.

```
b = a**2
print(b)
```

9.8596

Use hidden inline R to report Python results inline: $\sqrt{2} = 1.4142136$ by calling from python.

Recalling the variable a set above: $a = 3.14$.

5 Grapics are important

Here is a figure

```
x = linspace(0, 6*pi, 200)
plt.plot(x, sin(x))
plt.show()
```

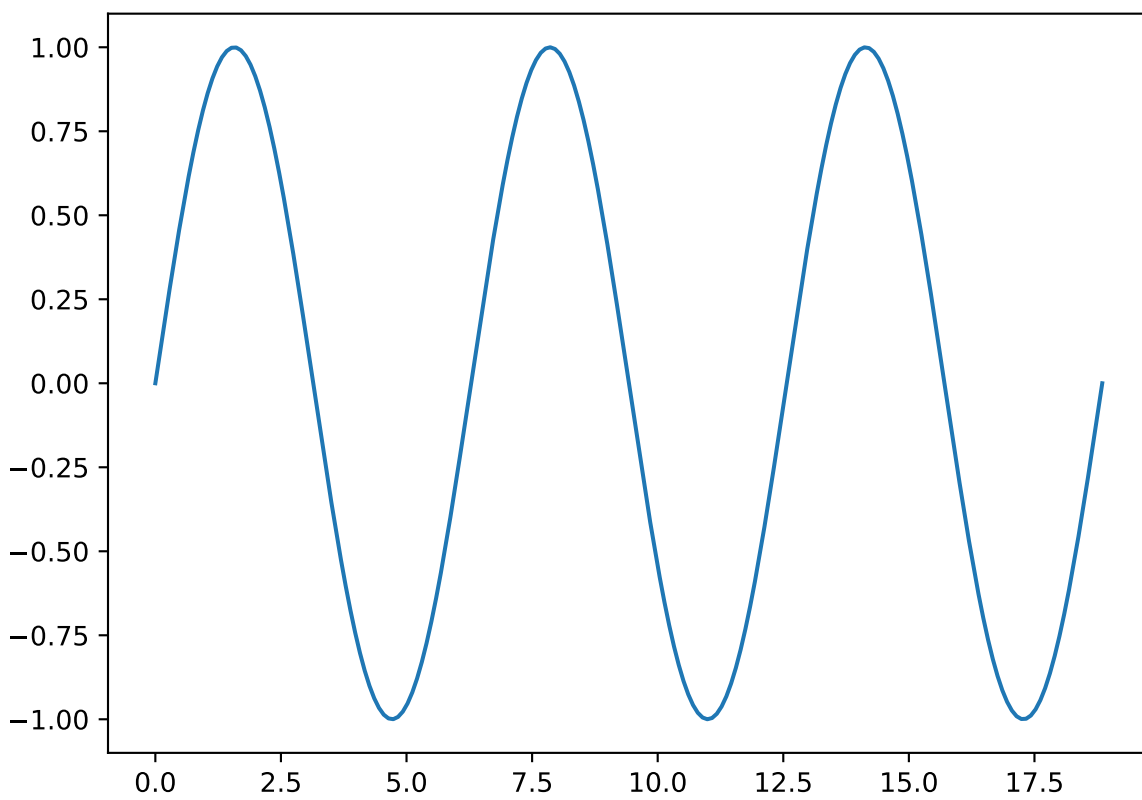


Figure 5.1: A plot of $\sin(x)$