IPython Notebooks and the Python Scientific Stack

IPython in action creating reproducible and publishable interactive work.

What is this?

This repo contains the full talk I intend to deliver (have delivered) at PyConZA2013. It contains all the files needed to build a final publishable PDF document and even adds a customer front page.

The Complete Talk GitHub Website can be accessed here

Background

IPython had become a popular choice for doing interactive scientific work. It extends the standard Python interpreter and adds many useful new futures. There is really no need to use the standard Python interpreter anymore. In addition to this IPython offers a web based Notebook that makes interactive work much easier, and have been used to write repeatable scientific papers and more recently a book has been written using this platform, the online Notebook Viewer and GitHub. The development of this material and tool chain to compile the notebook to a publishable PDF, has inspired me to maybe even try and turn this into a complete (free) book. Let's see what happens.

Combining the most common scientific packages with IPython makes it a formidable tool and serious competition to R. (R is still awesome!)

As a matter of fact you can run R in the notebook session, embed YouTube Videos, Images and lots more but let me not get ahead of myself....

The science stack consists of (but not limited to):

package	description
pandas	`dataframe` implementation (based on numpy)
scipy	efficient numerical routines
sympy	`symbolic mathematics`
matplotlib	python standard `plotting` package
sci-kit learn	machine learning and `well documented!`

Talk contents

The talk will aim to introduce these tools and give some practical examples. Once completed it will be shown how easy it is to publish your work to various formats. Some of the topics covered in the talk are listed below:

item	description
ipython	quick intro to ipython and the notebook
setup	set up your environment / get the talk files
notebook basics	navigate the notebook
notebook magic's	special notebook commands that can be very useful
getting input	as from IPython 1.00 getting input from sdtin is possible
local files	how to link to local files in the notebook directory
plotting	how to create beautiful inline plots

symbolic math	quick demo of sympy model
pandas	quick intro to pandas dataframe
typsetting	include markdown, Latex via MathJax
loading code	how to load a remote .py code file
gist	paste some of your work to gist for sharing
js	some javascript examples
customising	loading a customer css and custom matplotlib config file
git cell	add code to a special cell that would commit to git
output formats	how to publish your work to html, pdf or jeveal.js presentation

Get the talk files here

format	description
IPython notebook	.ipynb file to run in browser
IPython html notebook	converted to HTML and served online
IPython pdf notebook	converted to PDF for download (to be added, needs pandoc)
IPython pdf book	converted to pdf and a front-page stitched to it)
Ipython reveal.js presentation	converted to a reveal.js presentation and served online
Online IPython NBveiwer	view on the ipython notebook viewer

Dependencies

I was given the challenge to develop all of this on a Windows machine as some of my sponsors want to demonstrate that this stuff can not only be done on GNU/Linux/OSX. So all the tool chains are Windows based. If you know Linux, then you are the type of person that would easily port this. That being said the Windows GitHub client is refreshing.

package	description
IPython	To use NBConvert you need V1.00. If you only want to use the interactive notebook then v0.13 will be ok.
pandoc	The document converter used by IPythonr
MikeTex	If you want to do a TEX to PDF transform. I had so many issues with the TEX to PDF conversion by NBConvert, so settled for wkhtmltopdf(below) to convert HTML to PDF rather. (Convert notebook to HTML with NBconvert and then from HTML to PDF with wkhtmltopdf
wkhtmltopdf	Convert HTML to PDF
pdftk	Can be used to combine PDF's. In this case add a frontpage to the generated IPython notebook PDF.
ImageMagick	for compressing the PDF. Still experimenting with this.
GhostScript	needed by ImageMagick
anaconda	install anaconda from Continuum Analytics. Almost all the Python packages are included and it has a virtual environment manager via it's console application `conda'

How to run the Interactive Notebook

Navigate to the `src` directory and run from the command line:

```
python       ipython notebook
```

If everything works your browser should open and you can select the notebook and start experimenting!

PDF, HTML, Slideshow Build Script

There is a build script in the src directory. It is an IPython file. You can basically build shell scripts this way. To use the power of IPython commands save the file with the .ipy extension and call it with IPython. Even the magic's work. To build the document use ipython builddocs.ipy You will have to change the paths to the software however.

Cross Platform Output Rendering

I have tested the HTML outputs on my Galaxy S3 and S4, IPAD and Nexus7. They render very well. Even the downloaded PDF was easily readable on the NEXUS 7 in landscape mode. In conclusion the produces work is really very well packaged and easily consumed on most platforms. This is not bad, and all done with open source software.

Some interesting links

- A book written with IPython Notebook
- Notebook Viewer
- Anaconda - Installing almost everything you need

About the presenter

- I am an Electrical Engineer and is currently working for a consulting firm where I manage the Business Analytics and Quantitative Decision Support Services division.
- I use python in my day to day work as a practical alternative to the limitations of EXCEL in using large data sets.
- LinkedIn
- I am also a co-founder at House4Hack

------------------------------insert something cool here

IPython Notebook (A cookbook?)

Quick Ipython Introdution

Some Imports and Settings

```
%pylab --no-import-all inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

```
#Some standard stuff. Also see last cell for custom css
import json
s = json.load( open("static/matplotlibrc.json") )
matplotlib.rcParams.update(s)
figsize(18, 4)
print "Imported customer plotting"
```

Notebook Basics

Using Markdown

Notebook Magics

User Input

```
raw = raw_input("enter your input here >>> ")
print "hallo ", raw
```

```
enter your input here >>> tobie
hallo  tobie
```

Linking to Local Files

```
from IPython.display import FileLink, FileLinks
FileLinks('.')
```

```
./
  build.bat
  builddocs.ipy
  notebook.bat
  pycon13_ipython.ipynb
  README.md
  test.png
.\.ipynb_checkpoints/
  pycon13_ipython-checkpoint.ipynb
.\output/
  pycon13_ipython.html
  pycon13_ipython.slides.html
  pycon13_ipython_complete.pdf
  pycon13_ipython_pdf.pdf
.\static/
  custom.css
  frontpage.docx
  frontpage.pdf
  matplotlibrc.json
```

Adding Images

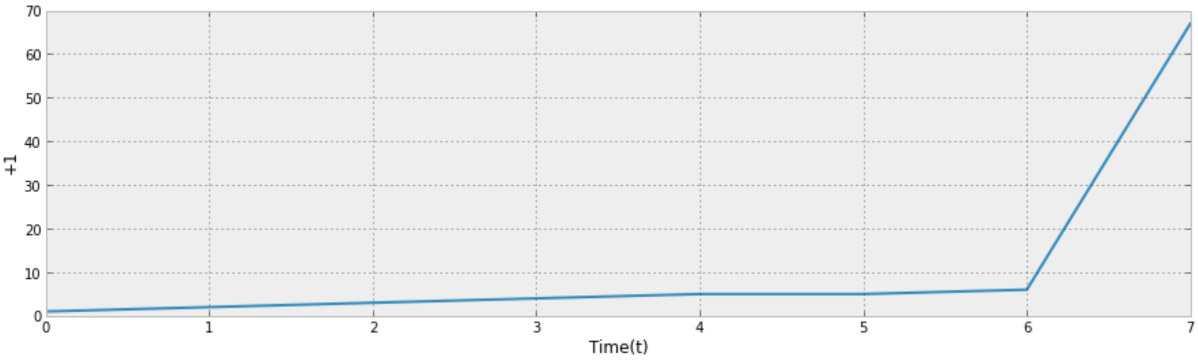
Adding Youtube Videos

```
from IPython.display import YouTubeVideo
YouTubeVideo('iwVvqwLDsJo', width=600, height=400)
```

Plotting with Matplotlib

```
figsize(15, 4)
xlabel('Time(t)')
ylabel('+1')
plot([1,2,3,4,5,5,6,67])
```

```
[<matplotlib.lines.Line2D at 0x5a1c550>]
```



Symbolic math

Pandas

Typesetting

Debugging

```
%pdb
```

```
Automatic pdb calling has been turned ON
```

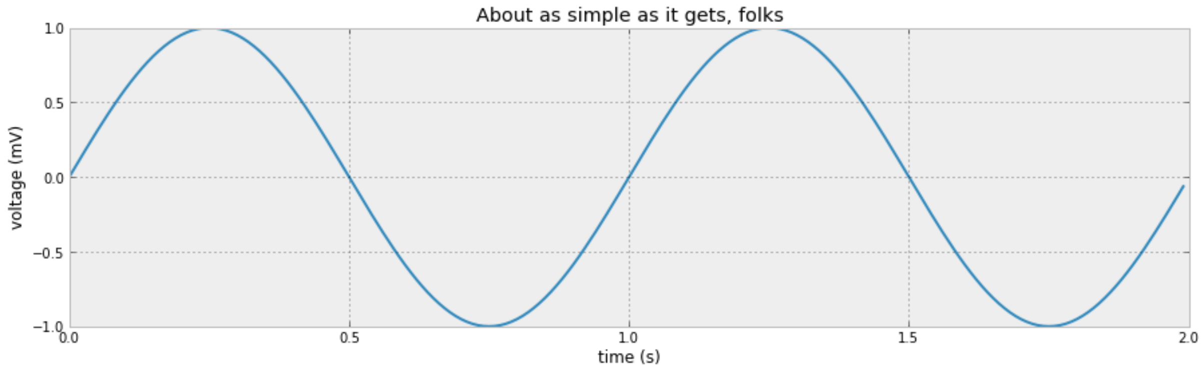
```
foo = 1
bar = 'a'
print foo+bar
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-4-08464a413a31> in <module>()
      1 foo = 1
      2 bar = 'a'
----> 3 print foo+bar

TypeError: unsupported operand type(s) for +: 'int' and 'str'
> <ipython-input-4-08464a413a31>(3)<module>()
      1 foo = 1
      2 bar = 'a'
----> 3 print foo+bar

ipdb> q
```

Loading Code Snippets

```
%load http://matplotlib.org/mpl_examples/pylab_examples/simple_plot.py
```

```
t = arange(0.0, 2.0, 0.01)
s = sin(2*pi*t)
plot(t, s)

xlabel('time (s)')
ylabel('voltage (mV)')
title('About as simple as it gets, folks')
grid(True)
savefig("test.png")
show()
```



Saving a Gist

```
%pastebin "cell one" 0-10
```

```
u'https://gist.github.com/6488549'
```

Publish Your Work

Changing the Notebook Layout

```
from IPython.core.display import HTML
def css_styling():

styles = open("static/custom.css", "r").read()
    return HTML(styles)
css_styling()
```

Some Interesting Notebooks