

Interactive Plotting in IPython Notebook (Part 2/2): Plotly

Summary

In [this previous post](#) I talked about interactive plotting packages that support the [IPython Notebook](#) and focused on [Bokeh](#). Today I'll be talking about [Plotly](#), a much richer package that allows for a lot more functionality. I will also provide some very rudimentary examples that should allow you to get started straight away.

Plotly

[Plotly](#) is in a lot of ways similar to [Bokeh](#) of which I've spoken in the [previous post](#) . However, I personally find it to be prettier, sexier, more convenient, offering much more functionality, and altogether a more well-rounded solution to your interactive plotting needs.

Not only does it offer a lot of 'basic' 2D plot types, while I know for a fact that the devs have been working on 3D plots as well, but it allows for real-time streaming data plots, online editing of your plots and data, a slew of APIs for different languages, and a whole wealth of features which you just can't find elsewhere.

On the other hand, [Plotly](#) isn't entirely free. Sure the basic free plan allows you to create as many plots as you want but they're all public and can be viewed by anyone. Should you want privacy, however, you gotta [pony up the dough](#), and at 12\$/month one might

find it kinda steep. In addition, unlike [Bokeh](#) , which serves the plots locally, [Plotly](#) does so through their own servers (unless you go for an enterprise-behind-my-own-firewall-solution) and needs to upload your data to do so. Thus, get ready for some notable lag.

Nonetheless, Plotly is a gorgeous lil' tool and IMHO in a league of its own among other tools, both in terms of features and ease of use.

Installation

At the time of writing, [Plotly](#) isn't hosted on the [conda](#) repos so one has to install it through the [pip](#) package manager simply with:

```
pip install plotly
```

Upon successful installation, the package should now be available for `import` for IPython console and [IPython Notebook](#) sessions.

The aforementioned conda repos can be found [here](#) for different platforms but one can easily use the `conda search` command and see whether a particular package, and which versions of said package, exists within the repos.

Setup

Before you start toying with [Plotly](#), you need to create a new account, which is free, and sign in.

Upon doing so, click on the upper-right corner at your username and go to 'Settings'. In the window that pops up, under the 'Profile' tab you will see two entries which you should take note of. The `username` and `API Key` are necessary to log onto [Plotly](#) from Python as we'll see in the [Usage section](#) below.

Usage

As an example I chose to simply redo the simple sine wave plot I showed on the [previous post](#) about Bokeh. After all, this post merely aims to showcase [Plotly](#) and as you'll see in the next section, the good [Plotly](#) folk have create a gorgeous and extensive doc-base.

Similarly to the [previous post](#) I once more assume that you're working directly in an IPython Notebook. This is actually important y'all as we have to use a different 'plotting function' as I'll explain below.

We simply begin by importing `plotly` and `numpy` to calculate and plot a sine wave as such:

```
import numpy
import plotly
```

Next, we need to sign in to Plotly using the `username` and `API Key` which you should have retrieved during the steps described in the [Setup section](#). Now in order to sign in you merely need a command such as the below:

```
plotly.plotly.sign_in("<enter your username here>", "<enter your API key here>")
```

At the time of writing the above function won't return an error upon failure to log in. However, when you try to plot something you will most likely get an `HTTPError: 500 Server Error: INTERNAL SERVER ERROR` error, so make sure your credentials are correct.

As in the [previous post](#), we use NumPy to calculate the 'coordinates' of a sine wave as such:

```
x = numpy.arange(0.0, 100.0, 0.1)
y = numpy.sin(x)
```

Now we just need to create a ‘trace’ as its called in the Plotly lingo using the `Scatter` function of the `graph_objs` module, passing the calculated coordinates:

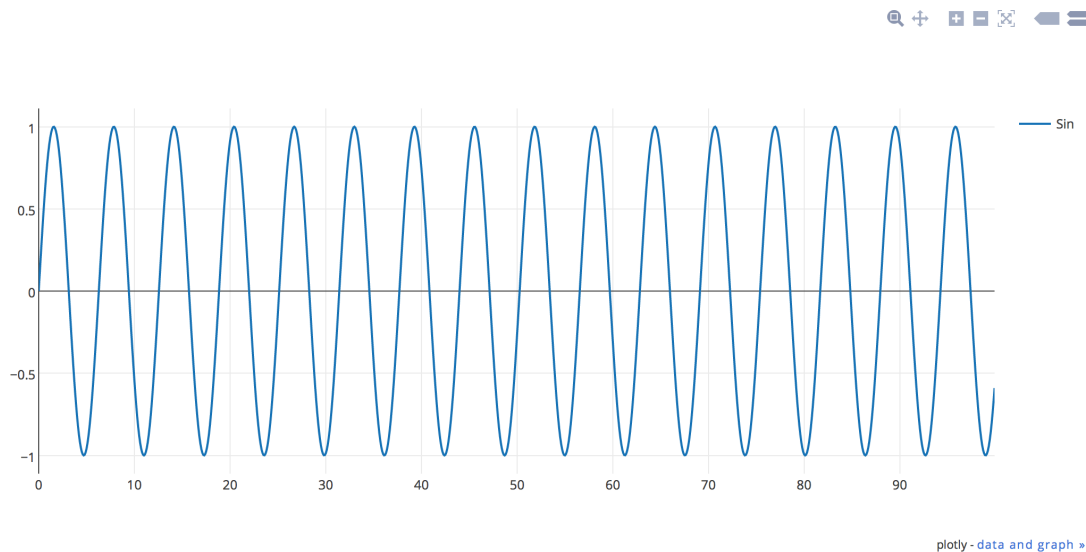
```
trace0 = plotly.graph_objs.Scatter(x=x,y=y, name="Sin")
```

The `graph_objs` module contains a tremendous variety of plot types. Visit the [Plotly Python API documentation](#) for a slew of examples.

Lastly, we pass a `list` of such traces to a one of the different ‘plot functions’ which reside under the `plotly.plotly` module. The one we need to use for an IPython Notebook is the `ipplot` function which will return an `IPython.core.display.HTML` object and force the resulting plot to appear as the output of the current cell. For this case the call is the following:

```
plotly.plotly.ipplot([trace0])
```

This last command will, after a few seconds, return an output that should look like the figure below. I’d like you to note the tools on the upper-right corner of the plot which allow for panning, zooming, and controlling the hover-tooltip behavior. In addition, by clicking on the `plotly - data and graph >>` link in the lower-right corner, you’ll be taken to the [Plotly website](#) where you can edit/format the plot, view/edit the data that generated, as well as share it with others.



The sine wave plot as generated by Plotly

Note that by using the `plotly.plotly.plot` function instead of `plotly.plotly.iplot` you will be taken directly to the [Plotly website](#) as if you clicked on the `plotly - data and graph >>` link.

You can find an IPython Notebook with the code used to create the above plot [here](#). Note that even though I've removed my credentials from the `plotly.plotly.sign_in` call, the plot should still show as its linking directly to that figure on the [Plotly website](#).

Resources

The [Plotly](#) people have created a beautiful [learn page](#) which has a ton of info including tutorials and guides.

Since here we're entirely focusing on Python, take a look at the corresponding [Getting Started](#) page and the [Plotly for Python User Guide](#). In addition, the [nbviewer](#) contains a [section](#) dedicated to [Plotly](#) which is pretty much a notebook'ed version of the aforementioned [User Guide](#).

Lastly, the [Plotly Python API documentation](#) has a slew of examples with the accompanying stand-alone source code which is generated explicitly for your username and contains the necessary key you need to log in.

