# jupyter_intro

October 11, 2017

## 1 Introduction to Jupyter (JUlia, PYThon and R) notebooks:

(many other programming languages including **Matlab** are supported as well)

### 1.1 1. Installation

#### 1.1.1 Conda

Install instructions are given for conda users. You can of course use other ways to install the mentioned packages. If you don't know what the `conda` command is:

> "Conda is an open source package management system and environment management system..."

For more information check out conda.pydata.org.

If you want to use `Python` and are not too famliar with it yet, the easiest way would be to install Anaconda (install instructions).

If you either have limited disk space or just want to get `conda` (e.g. if you are not planning to use Python for programming), install Miniconda.

Both, Anaconda and Miniconda, install `conda` and python on your system. Anaconda comes with an additional set of widely used Python packages, whereas in Miniconda all packages need to be installed individually using the `conda install` command.

One of the advantages of using `conda` is, that you can create multiple environments, which have different packages and/or `Python` versions installed and do not interfere with the `Python` possibly installed by your operating system.

#### 1.1.2 Jupyter

To install `Jupyter` with `conda`, open a terminal and type

```
conda install jupyter
```

Now navigate to the directory where you have the jupyter notebook file(s) (`.ipynb`) or where you want them to be saved

```
cd path/to/folder
```

To run Jupter notebooks, type

```
jupyter notebook
```

Now a browser window should open with all files in the current directory listed in it. Either open a file, or create a new notebook, by using the New button at the top right corner.

#### 1.1.3 Different Kernels

Python is installed by default and a Jupyter notebook using Python can be chosen from the New dropdown menu. But there are many other programming languages that can be used in an Jupyter notebook. E.g.:

**Jupyter with R** If you don't have R installed yet, you can do so by using conda:

```
conda install -c r r-essentials
```

If you already have R installed, you just need to install the IRkernel:

```
conda install -c ipython-notebook r-irkernel
```

Now you should be able to start a new R-notebook from the New dropdown menu in your Jupyter browser window.

**Jupyter with JULIA** You need Julia version 0.3 or higher. You can install it from here. Once you have installed Julia, run it and install IJulia from inside Julia with the Pkg.add("IJulia") command from the command line:

```
$ julia
julia> Pkg.add("IJulia")
```

If you don't have Jupyter or IPython installed, aboves command will automatically install Miniconda and Jupyter notebook. To install new Julia packages from inside Julia, you can then use conda. For more information checkout the IJulia github page. Now Julia notebooks should be available from the New dropdown menu in your Jupyter browser.

**Jupyter with MATLAB** You need to have MATLAB installed on your system. To make it available in Jupyter, you need to install pymatbridge and matlab_kernel, e.g. using conda:

```
conda install -c ioos pymatbridge
conda install -c ioos matlab_kernel
```

You may need to set the environmental variable MATLAB_EXECUTABLE to the path where your Matlab binary is installed. E.g. on a Linux system using bash terminal and having the Matlab binary available in /usr/bin/matlab, you can add the following line to your ~/.bashrc file:

```
export MATLAB_EXECUTABLE=/usr/bin/matlab
```

For more information check our the matlab_kernel github page.

**Example notebooks here**

## 1.2 2. This is a Markdown cell.

### 1.2.1 Wikipedia says:

> "Markdown is a lightweight markup language with **plain text formatting syntax** designed so that it can be converted to HTML and many other formats using a tool by the same name."

In `Jupyter` (and IPython) notebooks, we can even use LaTeX within Markdown cells. E.g. L'Hôpital's rule:

$$\lim_{x\to 0}\frac{e^x-1}{2x} \overset{\left[\frac{0}{0}\right]}{\underset{\mathrm{H}}{=}} \lim_{x\to 0}\frac{e^x}{2} = \frac{1}{2}$$

Check out e.g. this Markdown Cheatsheat.

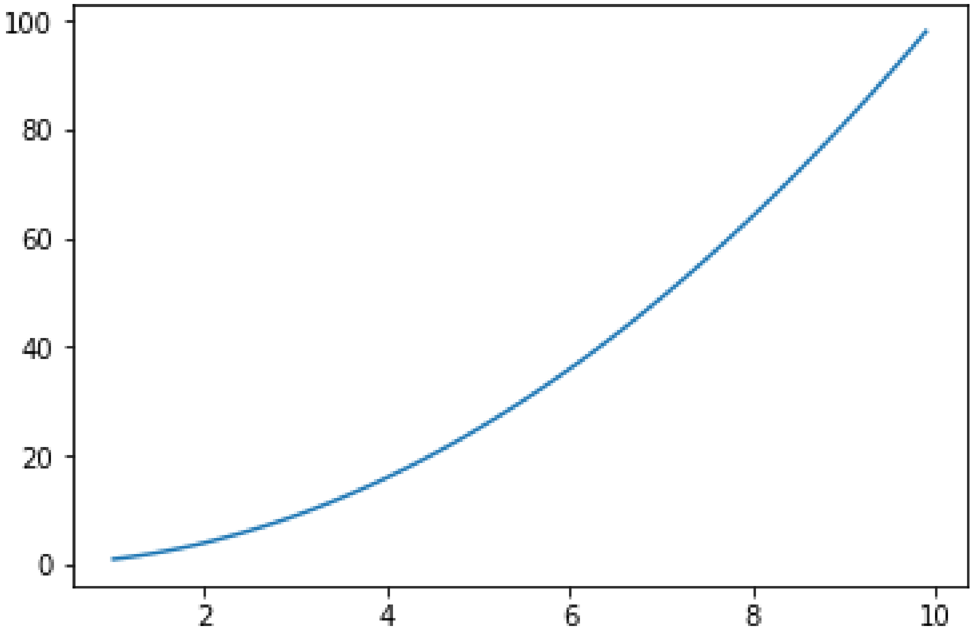
## 1.3 3. Inline Plotting

```
In [1]: import matplotlib.pyplot as plt
        import numpy as np

        X = np.arange(1,10, 0.1)
        plt.plot(X, X**2) # does not display the plot if inline is not active
```

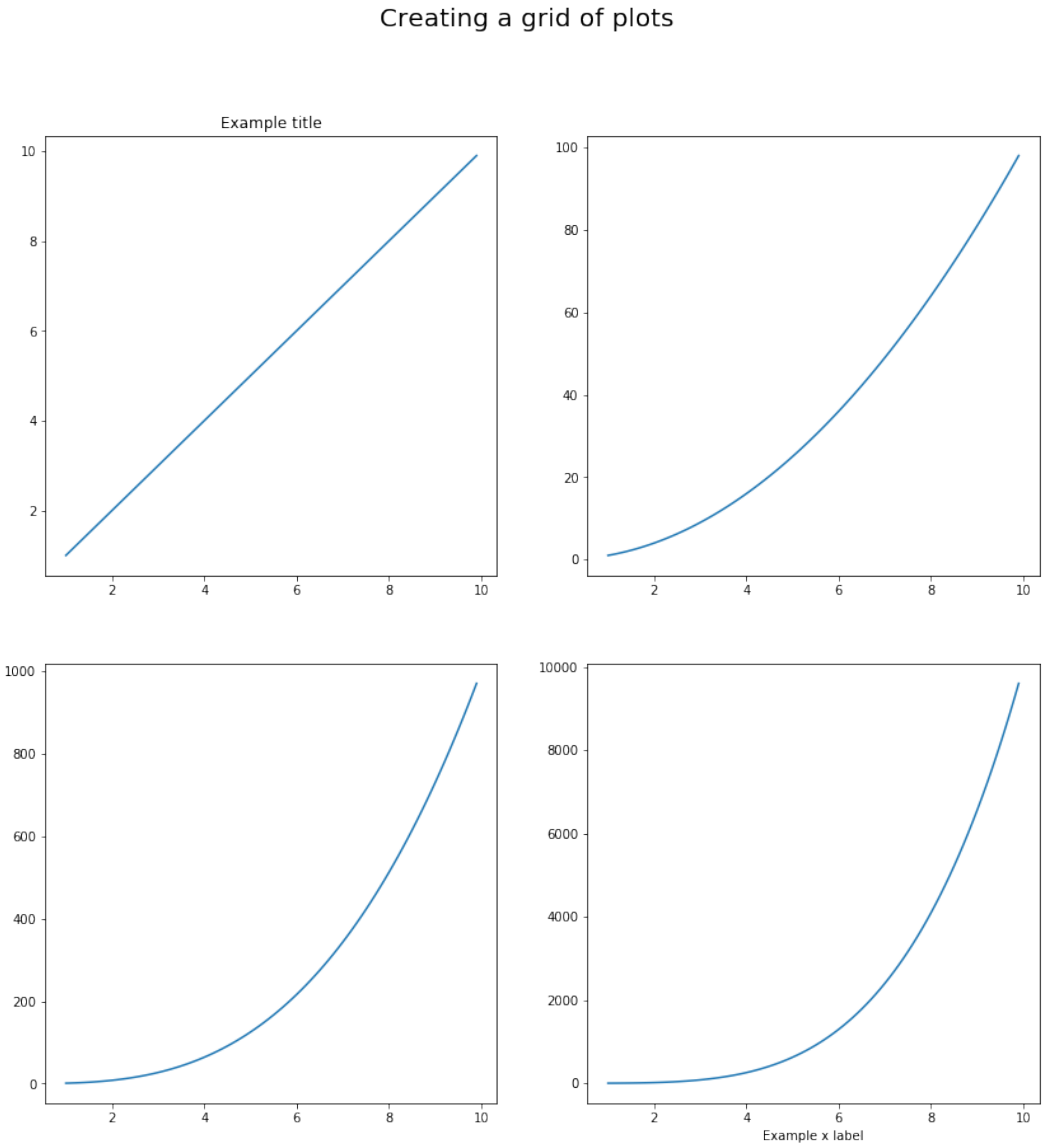
```
Out[1]: [<matplotlib.lines.Line2D at 0x7f5774641198>]
```

```
In [2]: # ipython magic command for displaying plots in the notebook
        % matplotlib inline

        plt.plot(X, X**2)
```

```
Out[2]: [<matplotlib.lines.Line2D at 0x7f57745382e8>]
```

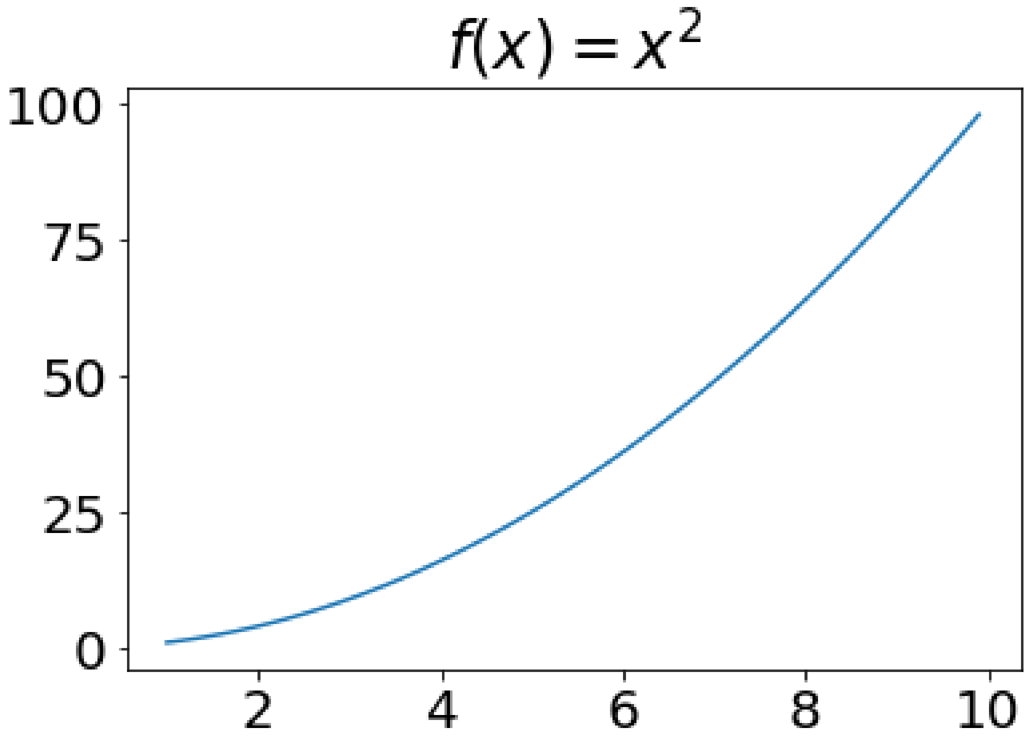


```
In [3]: fig, ax = plt.subplots(2, 2, figsize=(14,14))

        fig.suptitle('Creating a grid of plots', fontsize=20)

        ax[0,0].plot(X, X)
        ax[0,1].plot(X, X**2)
        ax[1,0].plot(X, X**3)
        ax[1,1].plot(X, X**4)

        ax[0,0].set_title('Example title')
        ax[1,1].set_xlabel('Example x label');
```



You can use LaTeX in figure labels and you can change the global fontsize for figures:

```
In [4]: plt.rcParams['font.size'] = 20
        plt.plot(X,X**2)
        plt.title(r'$f(x) = x^2$');
```



### 1.4 4. Unicode Identifiers

Works in Python3 and Julia. Just type e.g. `\beta + TAB` to get $\beta$ as variable.

```
In [5]:  = 5
         = 20
         =  /
        print()
```

```
4.0
```

### 1.5 5. Sharing Jupyter Notebooks

#### 1.5.1 Using Jupyter Drive (in Google Drive)

Check out the github page for more information.

#### 1.5.2 Using git

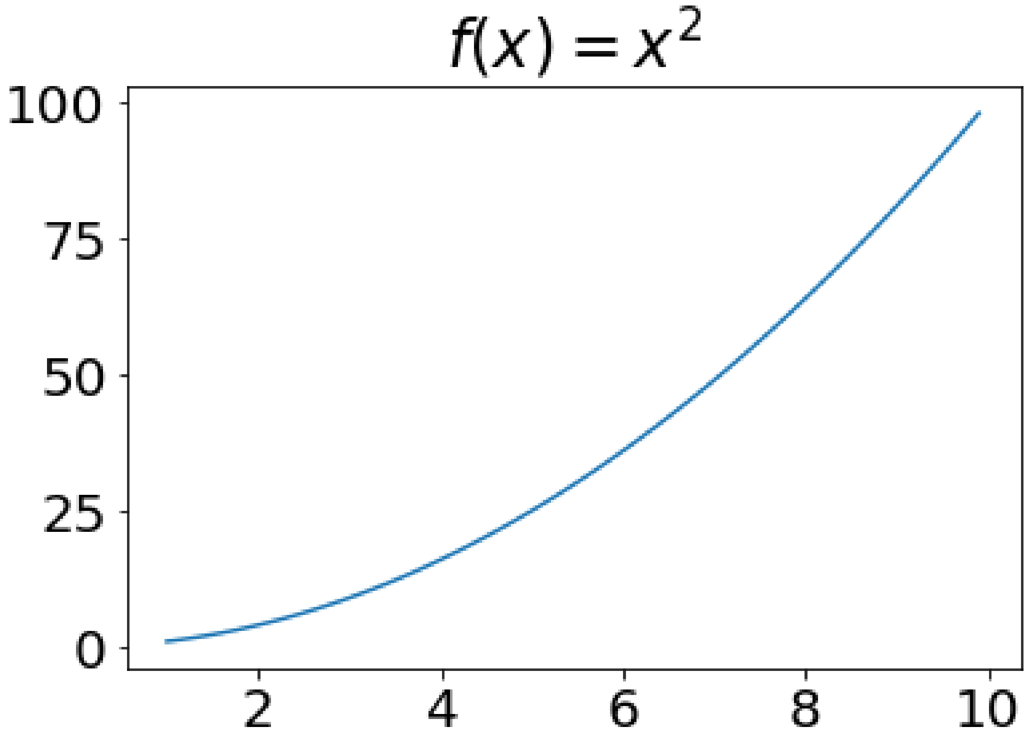
Problem: Notebook output get's indexed ⇒ annoying merge conlicts
Possible solution: e.g. this stackoverflow answer

## 1.6 6. Interactive Widgets

Note that it might be that on some systems for the interactivity to work the following command is required (before you start your notebook):

```
jupyter nbextension enable --py widgetsnbextension
```

```
In [6]: from ipywidgets import interact

        def f(n):
            plt.plot(X,X**n)
            plt.title(r'$f(x) = x^{}$'.format(n))
        interact(f, n=2);
```



## 1.7 7. For more information check out http://jupyter.org/