

[Sign in](#)[Get started](#)[Follow](#)

551K Followers

·

[Editors' Picks](#)[Features](#)[Explore](#)[Contribute](#)[About](#)

You have **2** free member-only stories left this month. [Sign up for Medium and get an extra one](#)

# 10 Useful Jupyter Notebook Extensions for a Data Scientist

These Jupyter Notebook Extensions make Data Scientist life easier



Sivasai Yadav Mudugandla · 1 day ago · 5 min read ★

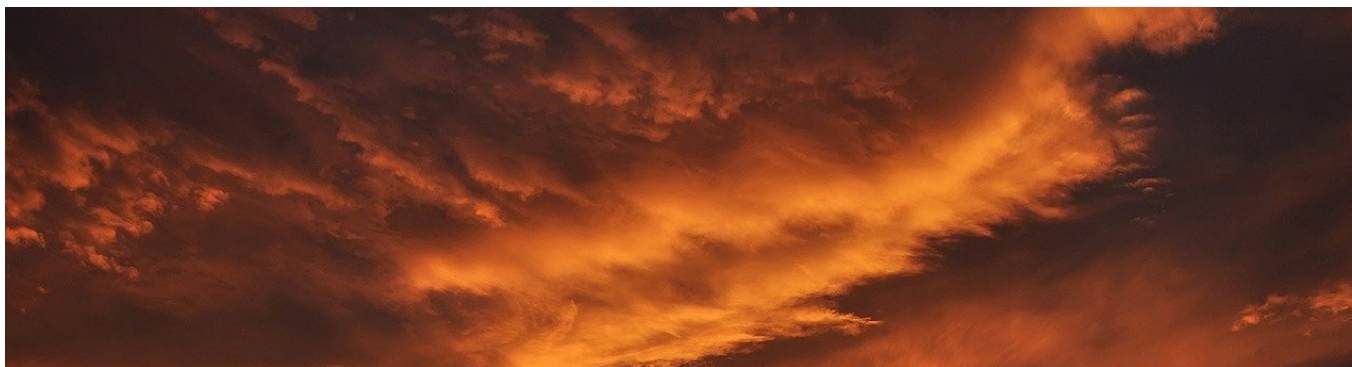




Image by [Bessi](#) from [Pixabay](#)

Every Data Scientist spends most of his time in data visualization, preprocessing and model tuning based on the results. These are the toughest situations for every Data Scientist because you will get a good model when you perform all these three steps precisely. There are 10 very helpful jupyter notebook extensions to help in these circumstances.

## 1. Qgrid

Qgrid is a Jupyter notebook widget which uses SlickGrid to render pandas DataFrames within a Jupyter notebook. This allows you to explore your DataFrames with intuitive scrolling, sorting and filtering controls, as well as edit your DataFrames by double-clicking cells.

Add Row		Remove Row						X
index	A	B	C	D	E			
0	2013-01-01	-2.0773	washington	foo				✓
1	2013-01-02	-0.55236	adams	bar				
2	2013-01-03	-1.46247	washington	buzz				
3	2013-01-04	0.85559	madison	bippity				
4	2013-01-05	-1.66466	lincoln	boppity				
5	2013-01-06	0.85659	jefferson	foo				✓
6	2013-01-07	1.23665	hamilton	foo				✓
7	2013-01-08	-0.36515	roosevelt	bar				
8	2013-01-09	0.89955	kennedy	zoo				

Out[3]:

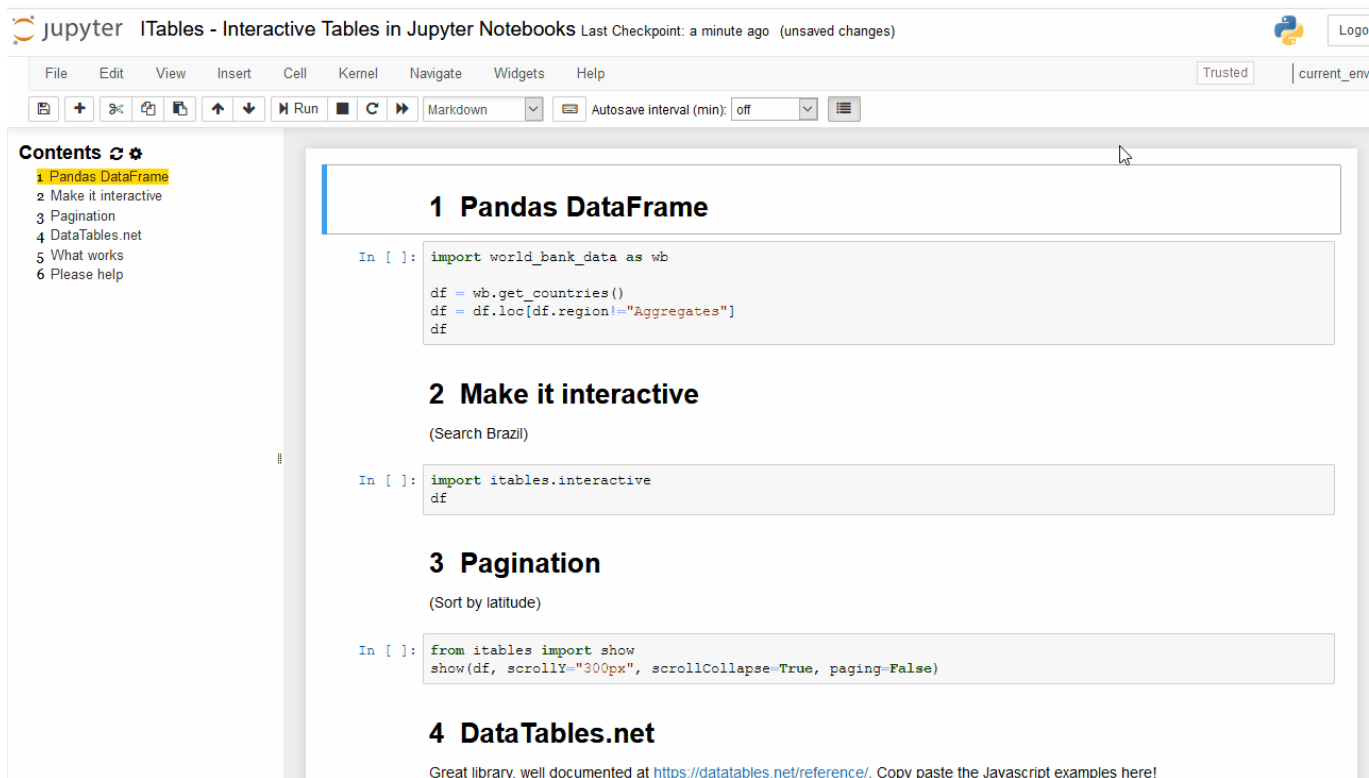
	A	B	C	D	E
0	2013-01-01	-2.077295	washington	foo	True
1	2013-01-02	-0.552359	adams	bar	False
2	2013-01-03	-1.462471	washington	buzz	False
3	2013-01-04	0.855593	madison	bippity	False
4	2013-01-05	-1.664660	lincoln	boppity	False
5	2013-01-06	0.856594	jefferson	foo	True
6	2013-01-07	1.236655	hamilton	foo	True
7	2013-01-08	-0.365152	roosevelt	bar	False
8	2013-01-09	0.899548	kennedy	zoo	False

```
pip install qgrid #Installing with pip
```

<https://towardsdatascience.com/10-useful-jupyter-notebook-extensions-for-a-data-scientist-bd4cb472c25e>

## 2. itables

ITables turns pandas DataFrames and Series into interactive data tables in both your notebooks and their HTML representation. ITables uses basic Javascript, and because of this, it will only work in Jupyter Notebook, not in JupyterLab.



The screenshot shows a Jupyter Notebook interface with the title 'jupyter ITables - Interactive Tables in Jupyter Notebooks'. The notebook contains four sections, each with a title and a code cell:

- 1 Pandas DataFrame**  
Code: 

```
In [ ]: import world_bank_data as wb  
df = wb.get_countries()  
df = df.loc[df.region!="Aggregates"]  
df
```
- 2 Make it interactive**  
(Search Brazil)  
Code: 

```
In [ ]: import itables.interactive  
df
```
- 3 Pagination**  
(Sort by latitude)  
Code: 

```
In [ ]: from itables import show  
show(df, scrollY="300px", scrollCollapse=True, paging=False)
```
- 4 DataTables.net**  
Great library, well documented at <https://datatables.net/reference/>. Copy paste the Javascript examples here!

Installation:

```
pip install itables
```

Activate the interactive mode for all series and dataframes with

```
from itables import init_notebook_mode
init_notebook_mode(all_interactive=True)

import world_bank_data as wb

df = wb.get_countries()
df
```

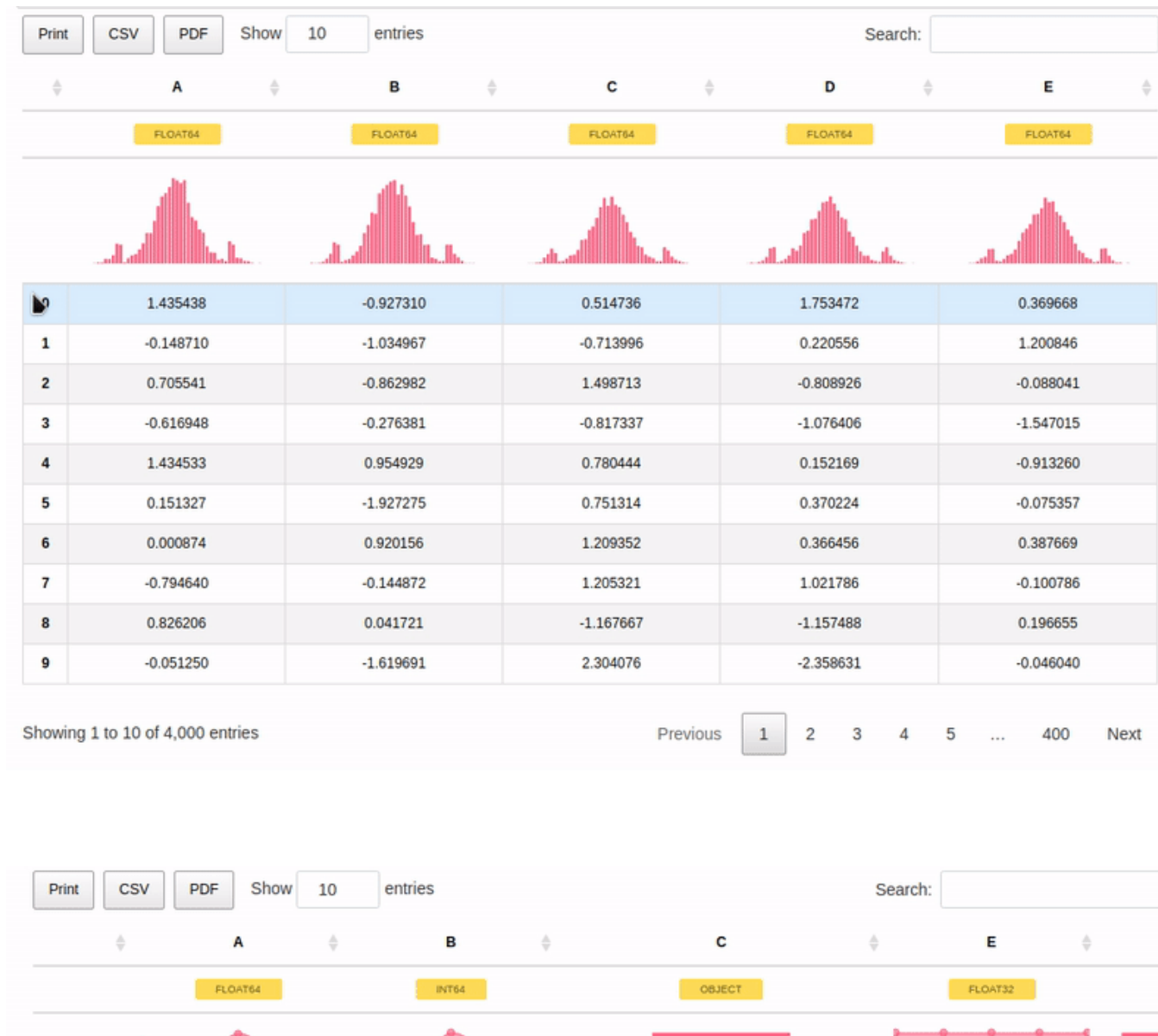
### 3. Jupyter DataTables

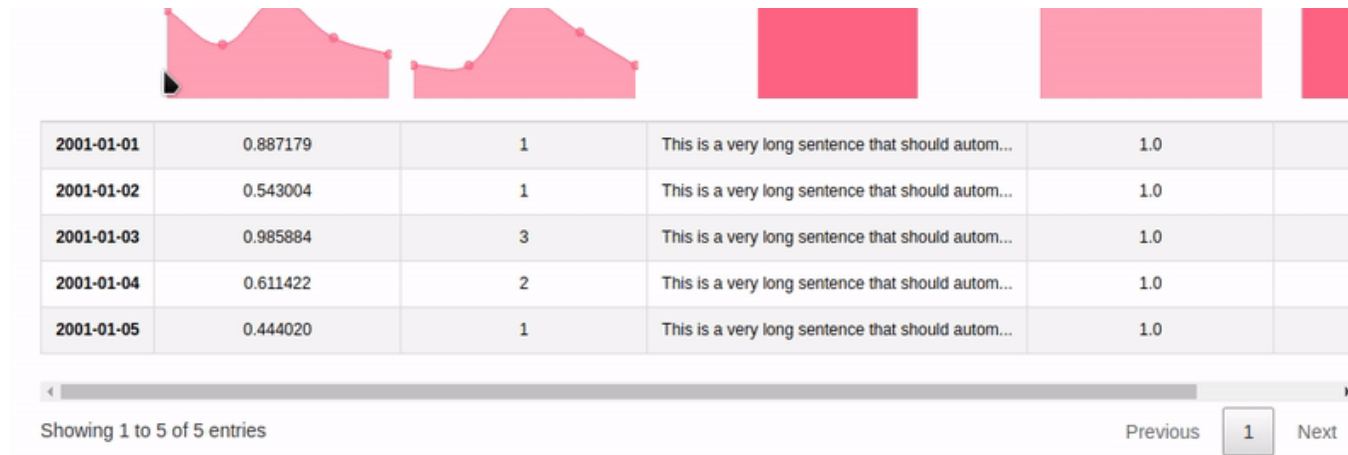
Data scientists and in fact many developers work with dataframe on daily basis to interpret data to process them. The common workflow is to display the dataframe, take a look at the data schema and then produce multiple plots to check the distribution of the data to have a clearer picture, perhaps search some data in the table, etc...

What if those distribution plots were part of the standard DataFrame and we had the ability to quickly search through the table with minimal effort?

## What if it was the default representation?

The `jupyter-datatables` uses `jupyter-require` to draw the table.





## Installation:

```
pip install jupyter-datatables
```

## Usage

```
from jupyter_datatables import init_datatables_mode  
init_datatables_mode()
```

## 4. ipyvolum

ipyvolume helps in 3d plotting for Python in the Jupyter notebook based on IPython widgets using WebGL.

Ipyvolume currently can

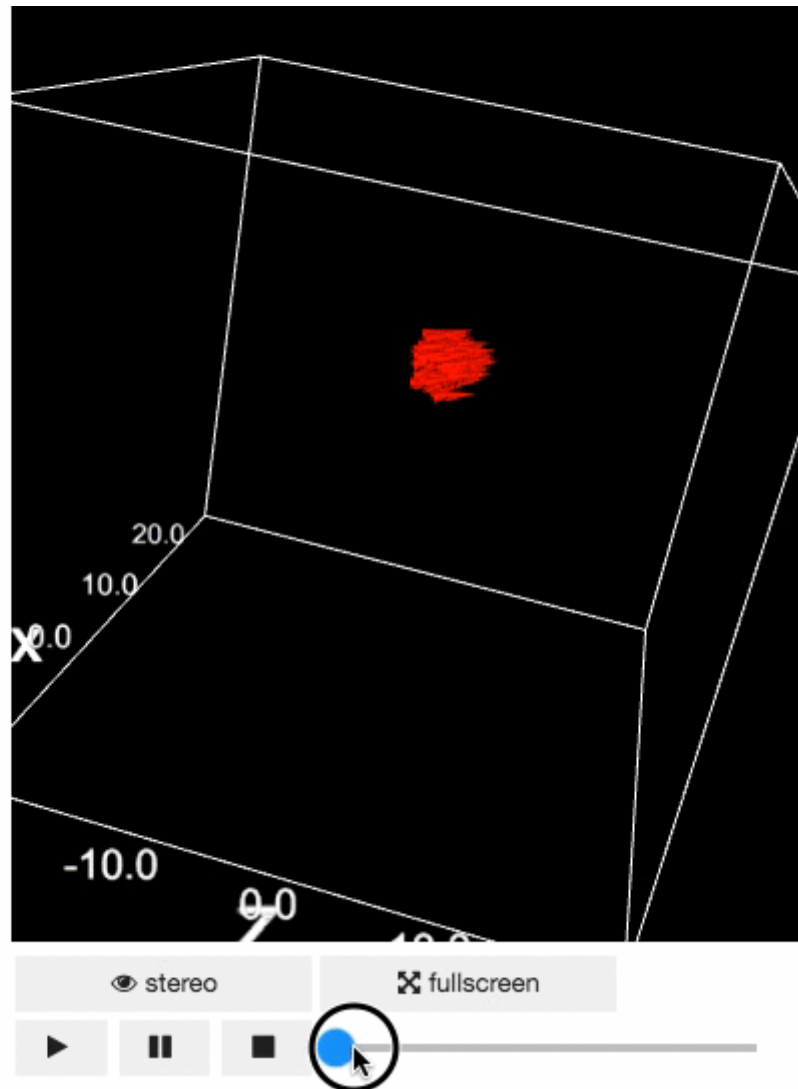
- Do (multi) volume rendering.
- Create scatter plots (up to ~1 million glyphs).
- Create quiver plots (like scatter, but with an arrow pointing in a particular direction).
- Do lasso mouse selections.
- Render in stereo, for virtual reality with Google Cardboard.
- Animate in d3 style, for instance, if the x coordinates or colour of a scatter plots changes.
- Animations / sequences, all scatter/quiver plot properties can be a list of arrays, which can represent time snapshots etc.

In [23]:

```
fig = p3.figure()
p3.style.use('dark')
s = p3.quiver(*ds_stream.data, size=6)
p3.animate_glyphs(s, interval=200)
p3.show()
```

×

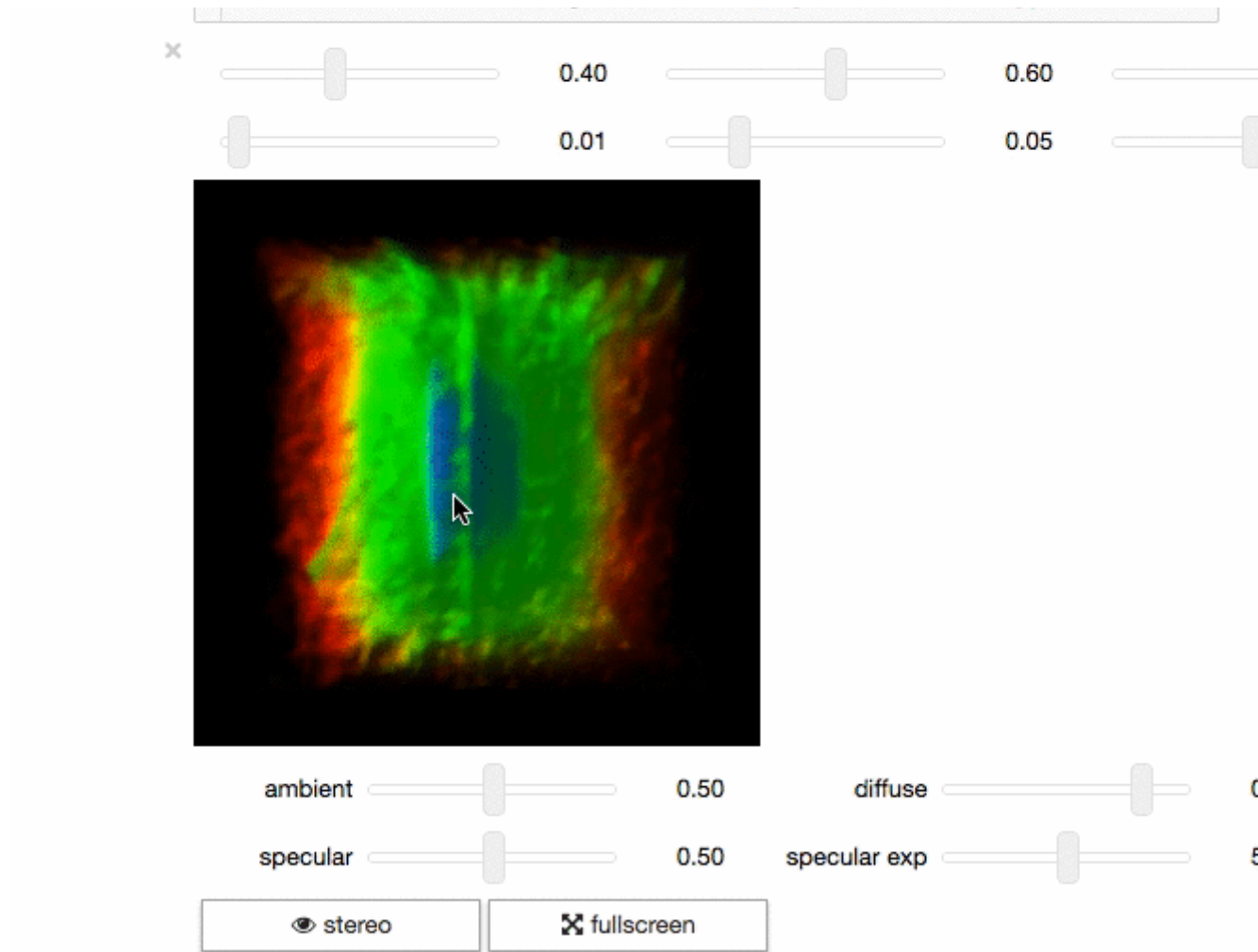




```
In [1]: import ipyvolume
```

```
In [2]: hdz = ipyvolume.datasets.hdz2000.fetch()
```

```
In [3]: ipyvolume.volshow(hdz.data, lighting=True, width=300,  
                           height=300, level=[0.4, 0.6, 0.9])
```



## Installation

```
$ pip install ipyvolume #Installing with pip
```

```
$ conda install -c conda-forge ipyvolume #Installing with conda
```

## 5. bqplot

bqplot is a 2-D visualization system for Jupyter, based on the constructs of the *Grammar of Graphics*.



## Goals

- provide a unified framework for 2-D visualizations with a pythonic API.

- provide a sensible API for adding user interactions (panning, zooming, selection, etc)

Two APIs are provided

- Users can build custom visualizations using the internal object model, which is inspired by the constructs of the Grammar of Graphics (figure, marks, axes, scales), and enrich their visualization with our Interaction Layer.
- Or they can use the context-based API similar to Matplotlib's pyplot, which provides sensible default choices for most parameters.

## Installation

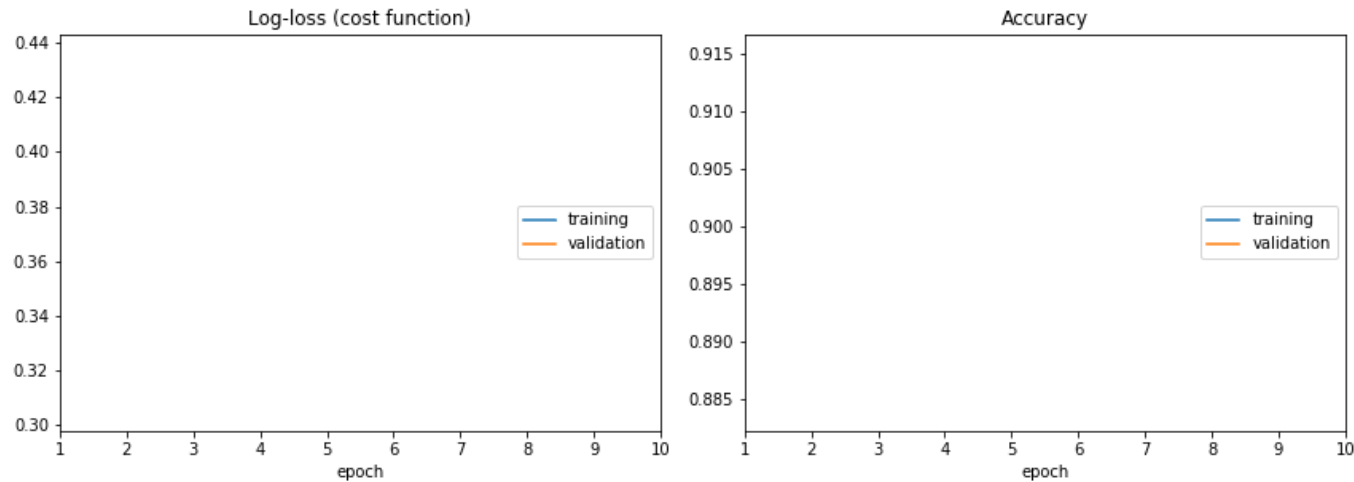
```
$ pip install bqplot #Installing with pip
```

```
$ conda install -c conda-forge bqplot #Installing with conda
```

## 6. livelossplot

Don't train deep learning models blindfolded! Be impatient and look at each epoch of your training!

liveplot provides a live training loss plot in Jupyter Notebook for Keras, PyTorch and other frameworks.



## Installation

```
pip install liveplot
```

## Usage

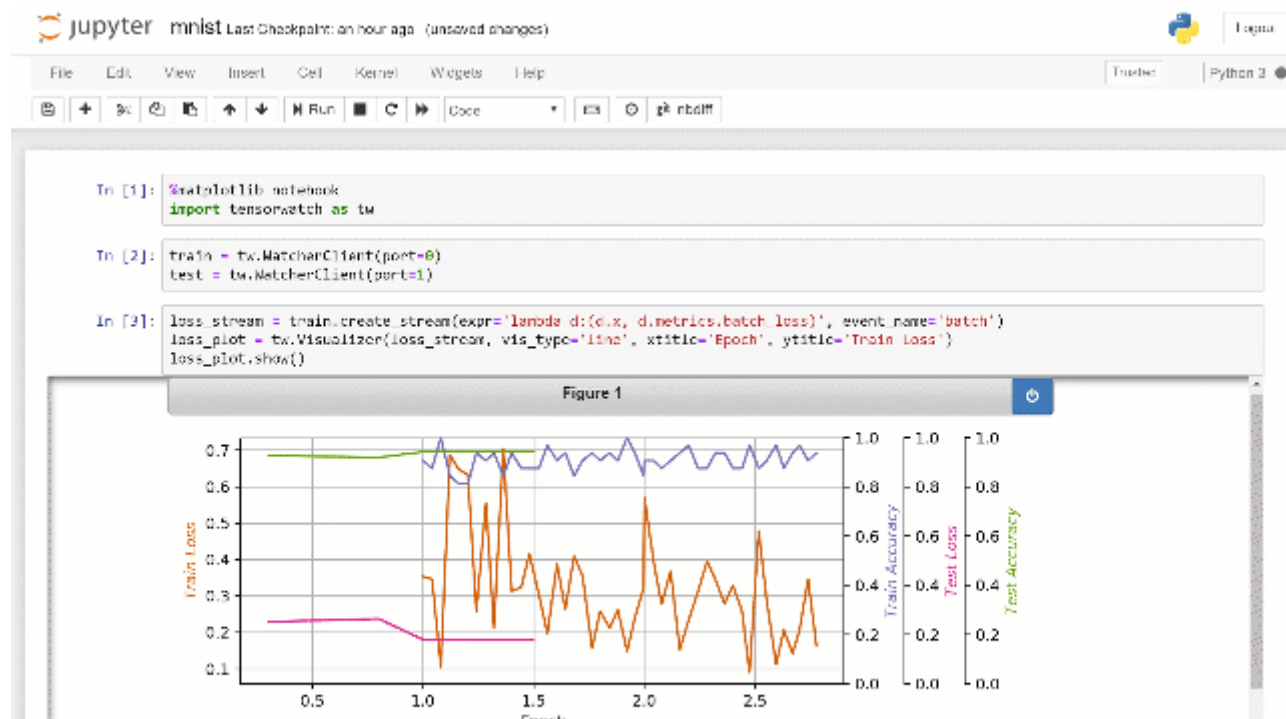
```
from liveplot import PlotLossesKeras

model.fit(X_train, Y_train,
          epochs=10,
```

```
validation_data=(X_test, Y_test),  
callbacks=[PlotLossesKeras()],  
verbose=0)
```

## 7. TensorWatch

TensorWatch is a debugging and visualization tool designed for data science, deep learning and reinforcement learning from Microsoft Research. It works in Jupyter Notebook to show real-time visualizations of your machine learning training and perform several other key analysis tasks for your models and data.



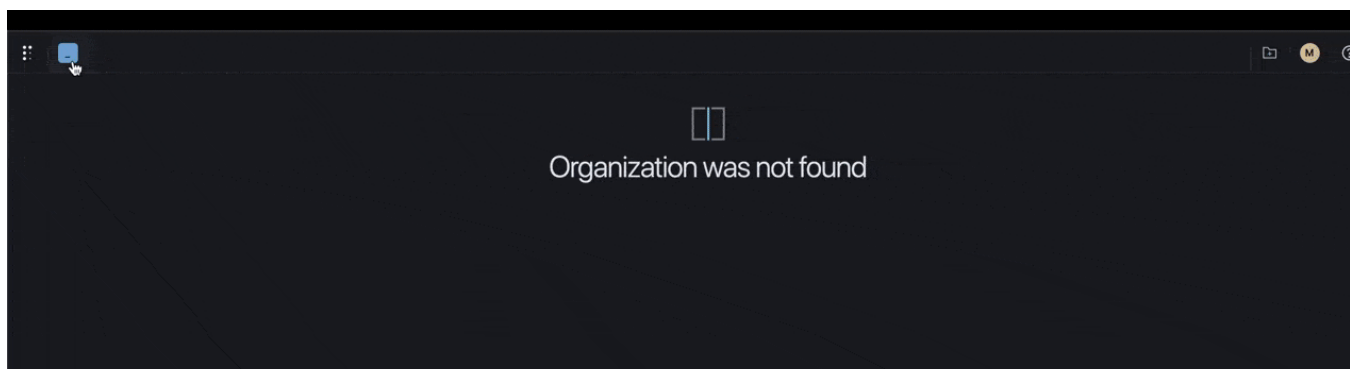
```
In [4]: acc_stream = train.create_stream(expr='lambda ds(d.x, d.metrics.batch_accuracy)', event_name='batch')
acc_plot = tv.Visualizer(acc_stream, vis_type='line', host=loss_plot, xtitle='Epoch', ytitle='Train Accuracy', yrange=(0,))
acc_plot.show()
```

## Installation

```
pip install tensorwatch
```

## 8. Polyaxon

Polyaxon is a platform for building, training, and monitoring large scale deep learning applications. We are making a system to solve reproducibility, automation, and scalability for machine learning applications. Polyaxon deploys into any data center, cloud provider, or can be hosted and managed by Polyaxon, and it supports all the major deep learning frameworks such as Tensorflow, MXNet, Caffe, Torch, etc.



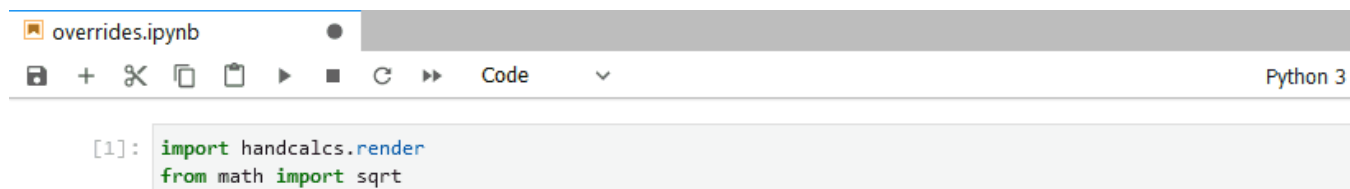


## Installation

```
$ pip install -U polyaxon
```

## 9. handcalcs

handcalcs is a library to render Python calculation code automatically in Latex, but in a manner that mimics how one might format their calculation if it were written with a pencil: write the symbolic formula, followed by numeric substitutions, and then the result.





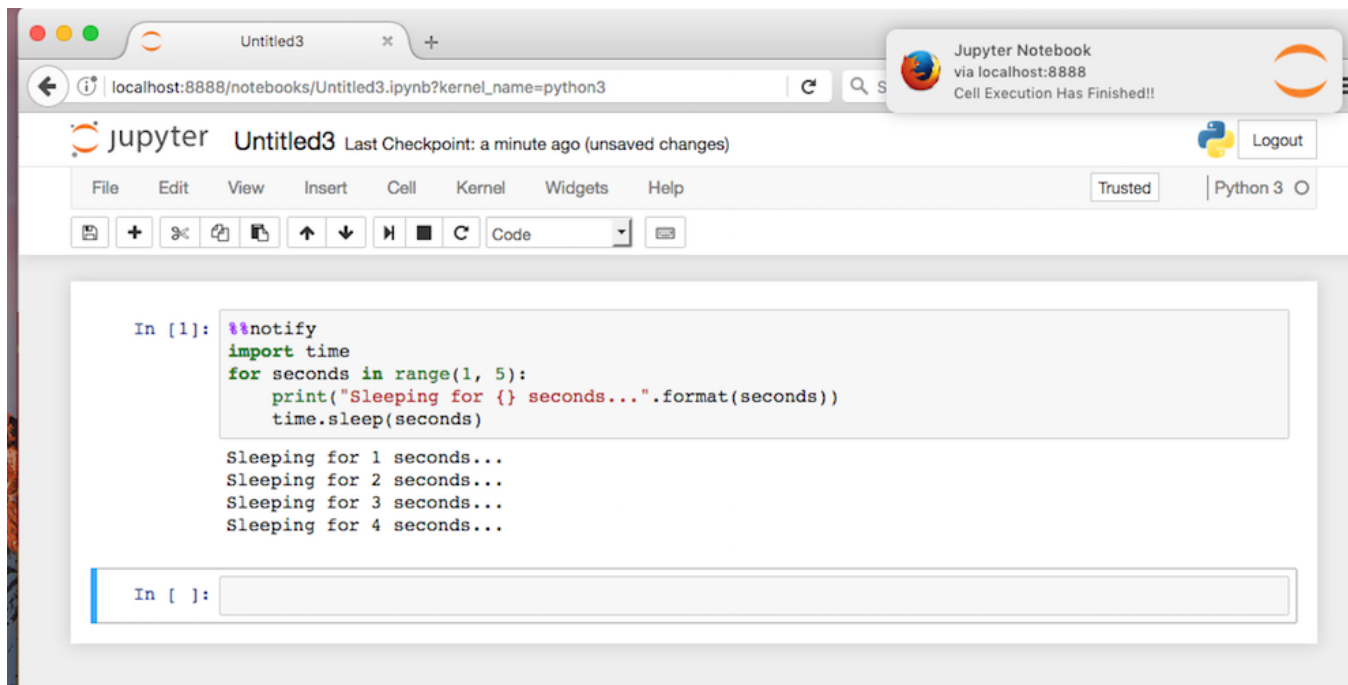


## Installation

```
pip install handcalcs
```

## 10. jupyternotify

jupyternotify provides a Jupyter notebook cell magic `%notify` that notifies the user upon completion of a potentially long-running cell via a browser push notification. Use cases include long-running machine learning models, grid searches, or Spark computations. This magic allows you to navigate away to other work and still get a notification when your cell completes.



## Installation

```
pip install jupyternotify
```

# Thank you for reading!

Any feedback and comments are, greatly appreciated!

---

## Sign up for The Daily Pick

By Towards Data Science

Hands-on real-world examples, research, tutorials, and cutting-edge techniques delivered Monday to Thursday. Make learning your daily ritual. [Take a look](#)

Your email

Get this newsletter

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.

Data Science

Jupyter Notebook

Machine Learning

Artificial Intelligence

Programming

