Beyond Jupyter: Other environments to do scientific programming in Python

Carlos Córdoba
ccordoba12@gmail.com

Universidad de los Andes / Quansight

SciPy Latin America
Bogotá, Colombia
October 10 2019

Outline

Outline

Who am I?

@ccordoba12

- **Spyder** maintainer
- **Software developer** at Quansight
- Former developer at **Anaconda** (3 years)
- **PhD Student** in Industrial Engineering at Universidad de los Andes, Colombia
- **Physicist** by training (Mathematica, C++)

Why Python became so relevant in science?

During the **last 6 years** we have witnessed the rise of:

- **Great libraries**:
  - **Pandas:** Excel on steroids
  - **Scikit-learn**: Machine learning for the masses
  - **Tensorflow** / **Pytorch**: Deep learning for the masses
- A **fantastic ecosystem** of tools to:
  - **Distribute**
  - **Install**
  - **Use**

Why Python became so relevant in science?

During the **last 6 years** we have witnessed the rise of:

- **Great libraries**:
  - **Pandas:** Excel on steroids
  - **Scikit-learn**: Machine learning for the masses
  - **Tensorflow** / **Pytorch**: Deep learning for the masses
- A **fantastic ecosystem** of tools to:
  - **Distribute**
  - **Install**
  - **Use**

Anaconda

A **unified platform** for science/engineering applications

$\Rightarrow$ Travis Oliphant and Peter Wang

Conda

Multi-platform general purpose **package manager** for **scientific packages**

⇒ Kale Franz, Ilan Schnell, Aaron Meurer

Conda-build

CONDA-build

A **build tool** that gets **scientific packages**

⇒ Ilan Schnell, Mike Sarahan and Ray Donelly

Conda-forge

CONDA-FORGE

Community packages by the thousands

⇒ Filipe Fernandes, John Kirkham, Jonathan Helmus

Mature scientific GUIs

Fernando Perez
Brian Granger

Pierre Raybaut
Me

Programming environments for **scientists/engineers**

Design philosophies

- Data **exploration**
- Computational **narratives**

- Focused on **programming** scientific code
- Provide **integrated** environment to do it

Outline

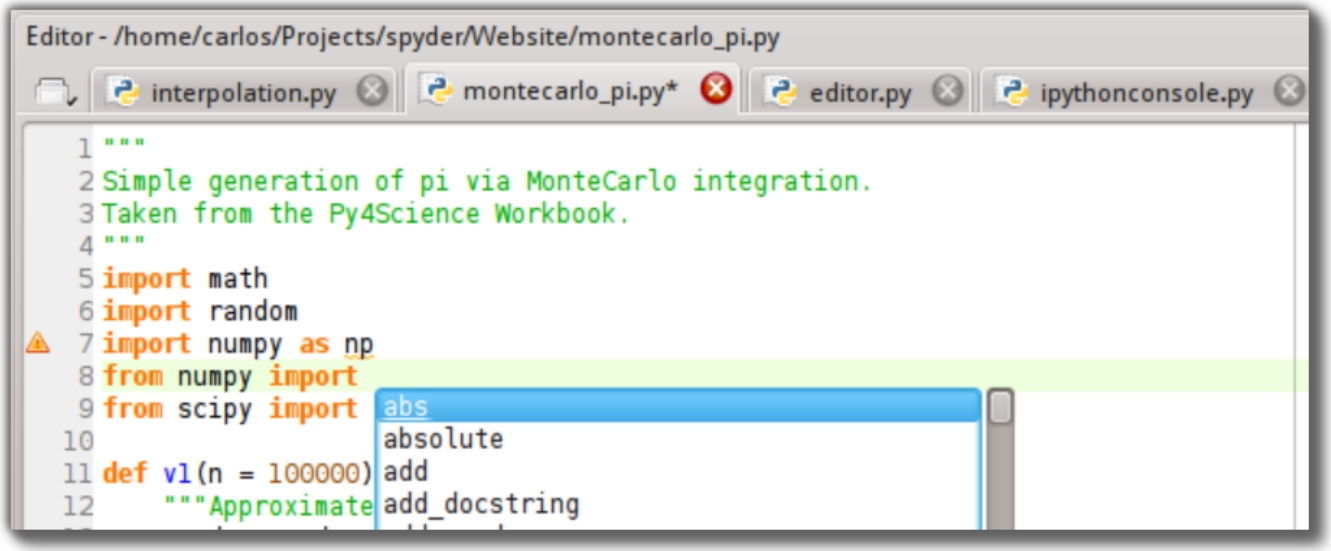
Some data about Spyder

- **Spyder** = The Scientific Python Development Environment
- Created by **Pierre Raybaut** in **2009**
- **MIT** licensed
- **Multi-platform** (Windows, macOS, Linux)
- **72.000** lines of code
- **100** contributors
- **50** pull requests / **200** issues **per month**

More about Spyder

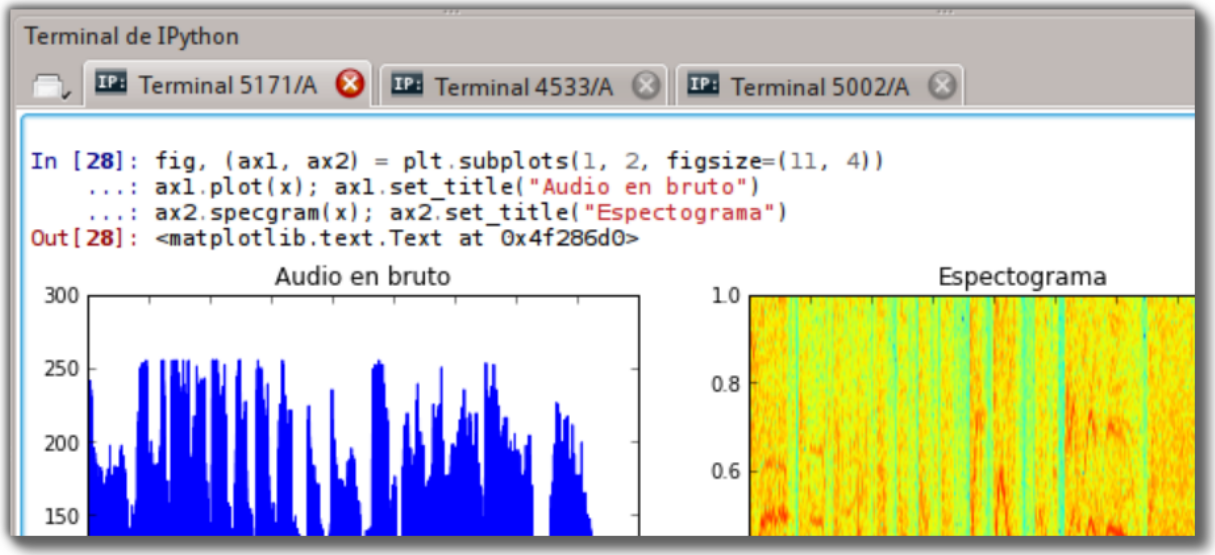
- **Website**: `https://www.spyder-ide.org/`
- **Github**: `https://github.com/spyder-ide/spyder`
- **9** core developers
- **2 million** downloads on Anaconda
- **350.000** downloads per year on PyPI

Our cornerstone: the editor



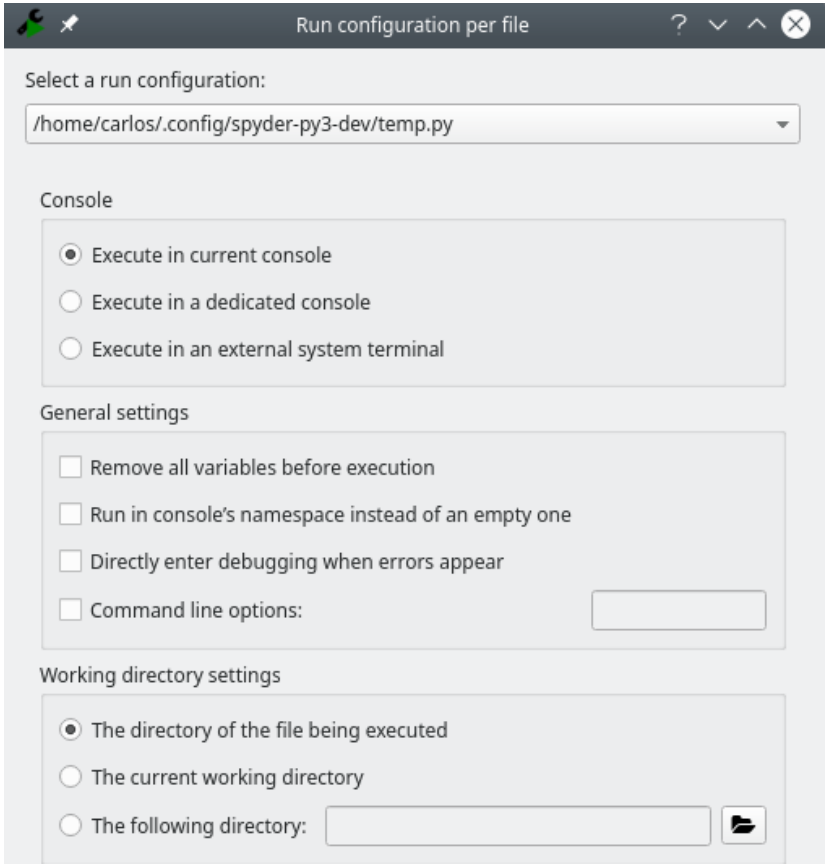
- **Code Completion** $\Longrightarrow$ Tab
- Highlight **errors** (red) and **warnings** (orange)
- Access to **docs** $\Longrightarrow$ Ctrl + I
- Go to **definition** $\Longrightarrow$ Ctrl + G
- **Multi-language** in version 4

Run your code in the console



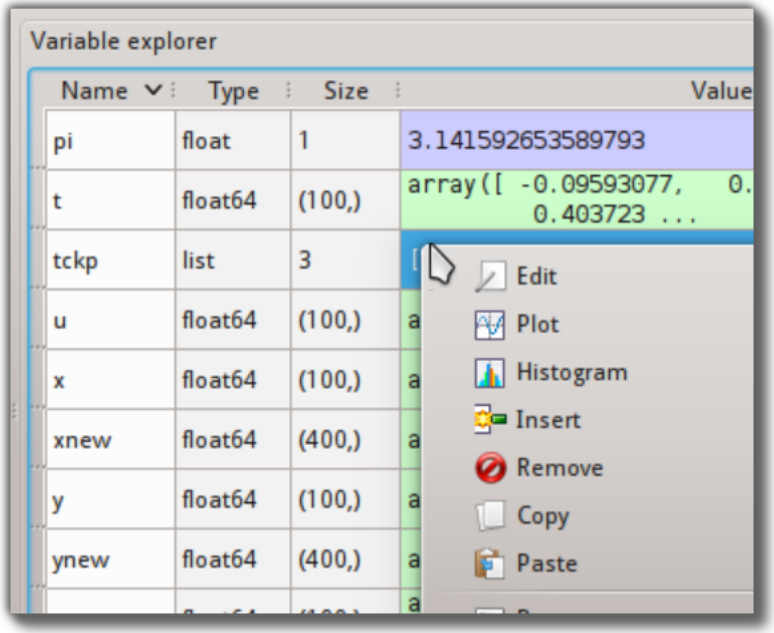
- As **many consoles** as you want!
- **One console per file** (if you want)
- **Special consoles** for SymPy, Cython and Pylab.
- Connect to **remote kernels**
- Access to **docs** $\Longrightarrow$ Ctrl + I

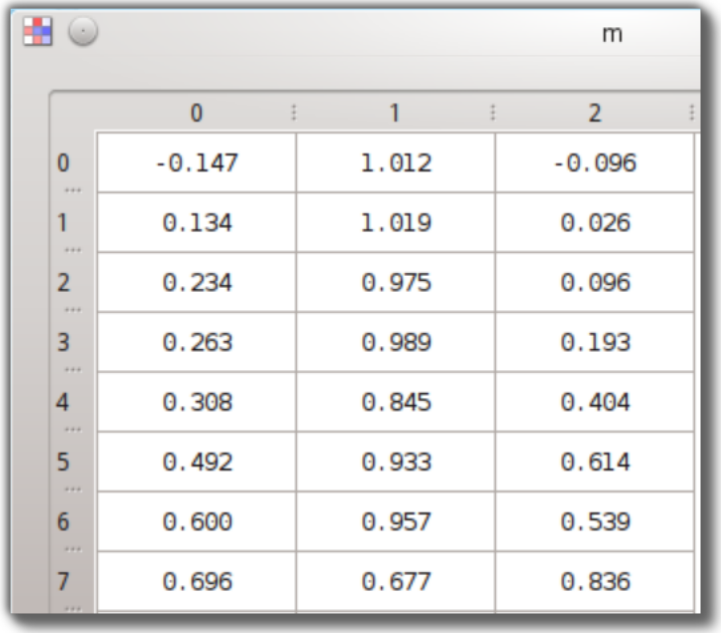
Multiple evaluation modes



- Run **file** $\Longrightarrow$ F5
- Run **cells** (# %%) $\Longrightarrow$ Ctrl + Enter or Shift + Enter
- Run **selection** or **line** $\Longrightarrow$ F9

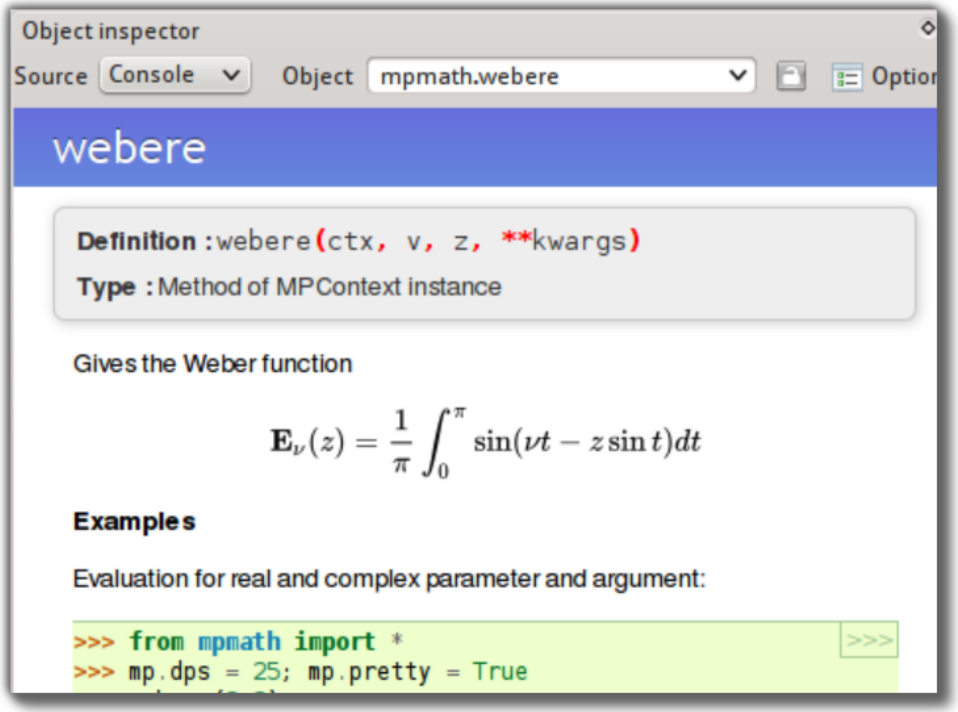
Variable Explorer





- **Inspect** variables defined in the console
- **Modify** their values **graphically**
- **Copy**, **plot** and **remove** variables
- Works while **debugging**!

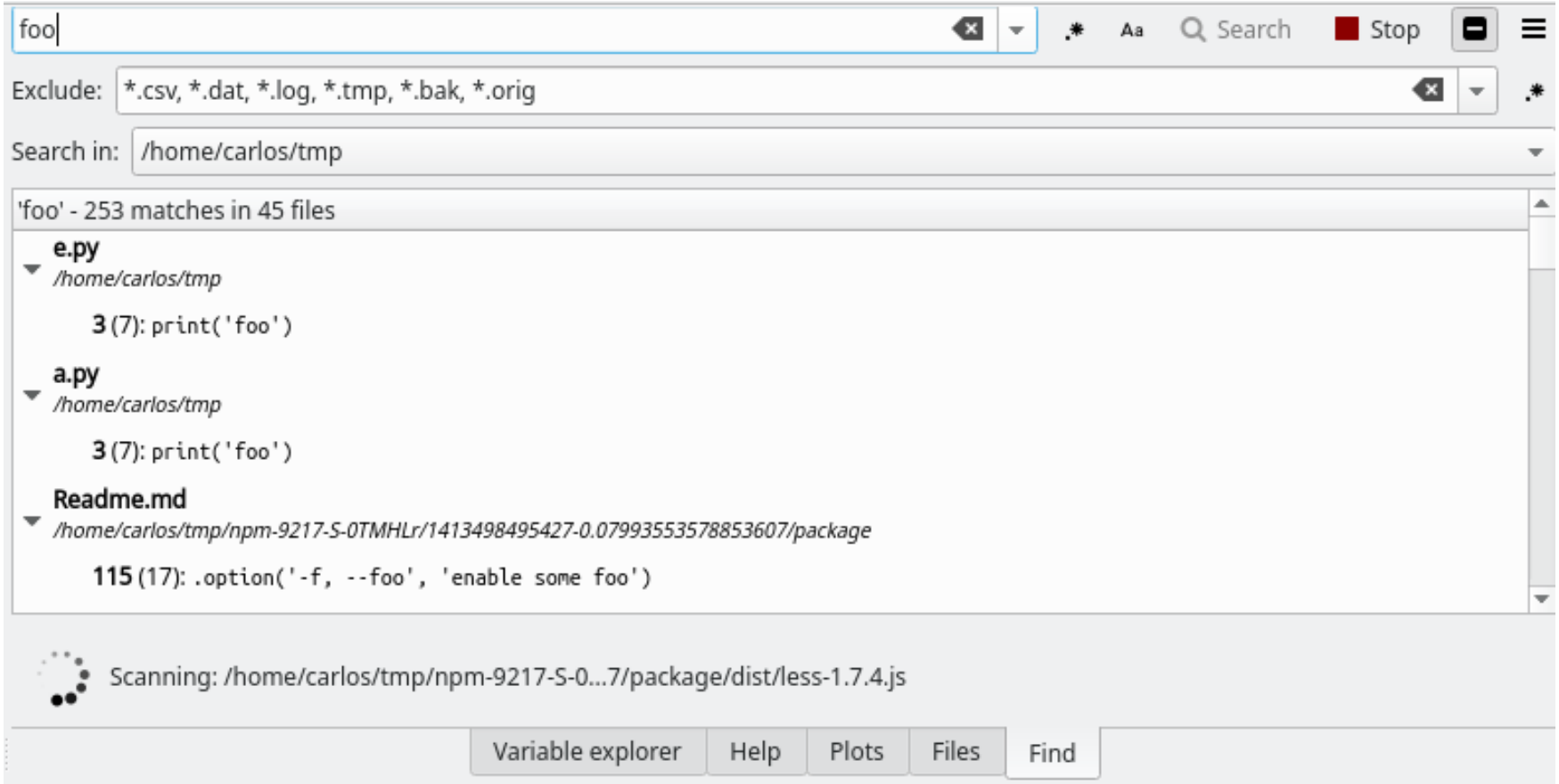
Help



- **Rich text** docstrings
- **Render math** equations written in Latex

Outline

Find in files



Projects

/home/carlos/tmp/qrs_test.py

- tmp
  - npm-9217-S-0TMHLr
  - test
  - 01. Numeros y listas.ipynb
  - a.py
  - bower.json
  - crox.csv
  - e.py
  - foo.ipynb
  - foo.py
  - foo.pyc
  - qrs_test.py
  - spyder_crash.log
  - test-mayavi.py
  - test.spydata
  - tt.txt
  - Untitled.ipynb

qrs_test.py

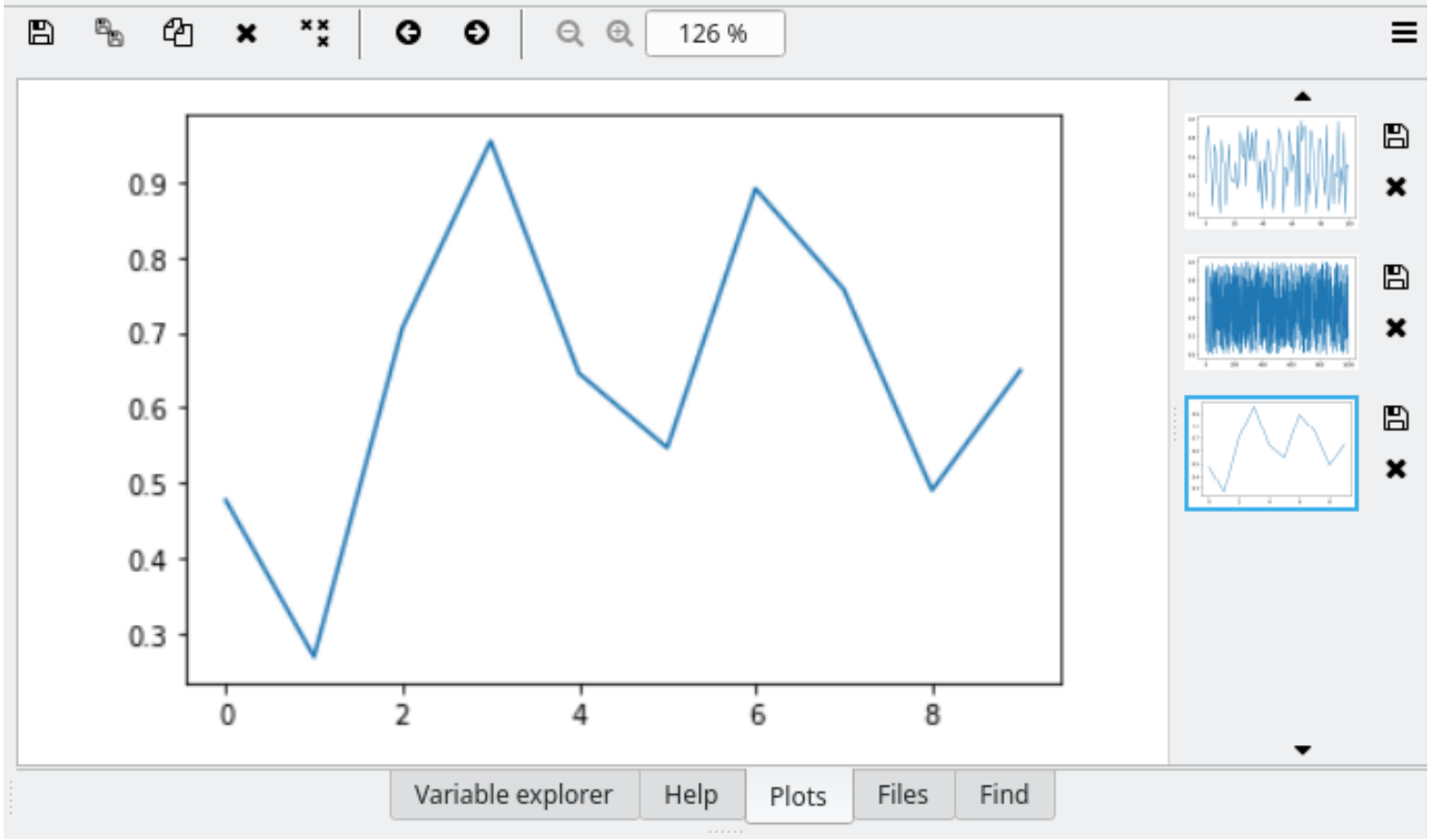
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Jan 11 15:30:03 2019

@author: mariano
"""

# Importo TensorFlow como tf
#import tensorflow as tf
# Importo keras
#import keras as kr

# Librerias auxiliares
import numpy as np
import matplotlib.pyplot as plt
#from pandas import DataFrame
#from IPython.display import HTML
import os
from glob import glob
import h5py
import wfdb as wf
from scipy import signal as sig
```

Plots


126 %
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
2
4
6
8
Variable explorer
Help
Plots
Files
Find

Outline

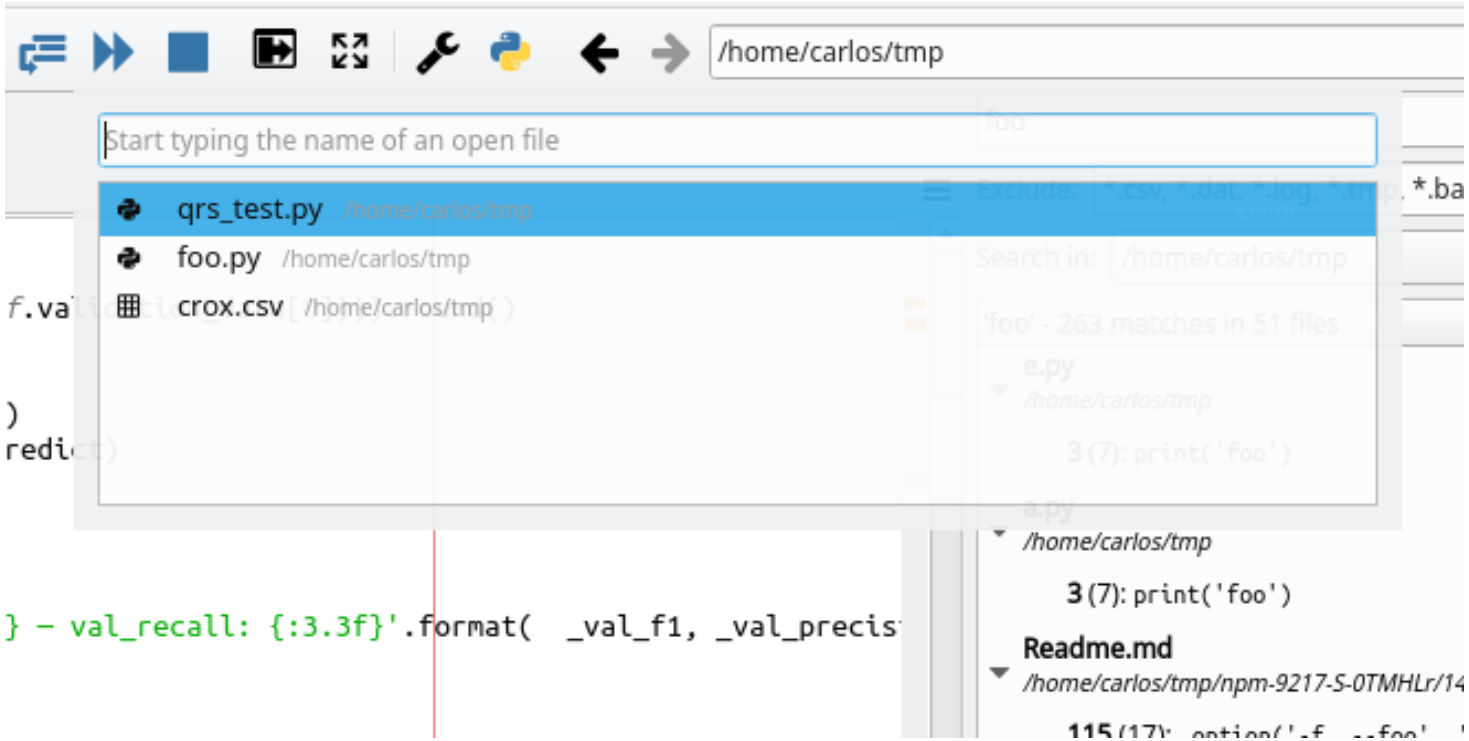
/home/carlos/tmp/qrs_test.py

- qrs_test.py
  - MyCallbackClass
    - on_train_begin
    - on_epoch_end
  - Sensitivity
    - se
  - Precision
    - pp
    - f1
    - get_records
    - get_beats
    - gen_interest_ranges
    - my_int
    - make_dataset

qrs_test.py

```
62
63      def on_epoch_end(self, epoch, logs={}):
64       val_predict = (np.asarray(self.model.predict
65       val_targ = self.validation_data[1]
66       _val_f1 = f1_score(val_targ, val_predict)
67       _val_recall = recall_score(val_targ, val_pre
68       _val_precision = precision_score(val_targ, v
69
70       self.val_f1s.append(_val_f1)
71       self.val_recalls.append(_val_recall)
72       self.val_precisions.append(_val_precision)
73       print('\nval_f1: {:3.3f} — val_precision: {:
74
75       return
76
77  # %% Sensitivity
78  def se(y_true, y_pred):
79      """Recall or sensitivity metric.
80
81      Only computes a batch-wise average of recall.
82
83      Computes the recall, a metric for multi-label
84      how many relevant items are selected.
85      """
86      true_positives = K.sum(K.round(K.clip(y_true
87      possible_positives = K.sum(K.round(K.clip(y_t
88      recall = true_positives / (possible_positives
89      return recall
90
91
92  # %%% Precision
93  def pp(y_true, y_pred):
94      """Precision or Positive Predictive Value met
95
```

Switcher


/home/carlos/tmp
Start typing the name of an open file
qrs_test.py /home/carlos/tmp
foo.py /home/carlos/tmp
crox.csv /home/carlos/tmp
} - val_recall: {:3.3f}'.format(_val_f1, _val_precis
/home/carlos/tmp
3 (7): print('foo')
Readme.md
/home/carlos/tmp/npm-9217-S-0TMHLr/14

Outline

Spyder

Advantages

- Integrated experience
- Powerful, but simple to use
- Translated interface to eight languages

Disadvantages

- Lots of installation issues
- Not easily extensible
- Crappy debugger

Spyder

Advantages

- Integrated experience
- Powerful, but simple to use
- Translated interface to eight languages

Disadvantages

- Lots of installation issues
- Not easily extensible
- Crappy debugger

Jupyter

Advantages

- Revolutionary way to let users mix text, code and graphics
- Clean, simple interface
- Multi-platform through the browser
- JupyterHub and Binder

Disadavantages

- Poor integration with the operating system
- Important functionality is left for extensions

Jupyter

Advantages

- Revolutionary way to let users mix text, code and graphics
- Clean, simple interface
- Multi-platform through the browser
- JupyterHub and Binder

Disadavantages

- Poor integration with the operating system
- Important functionality is left for extensions

Thanks

Thanks!

Correo: **ccordoba12@gmail.com**

Github: **@ccordoba12**

Twitter: **@ccordoba12**

`https://github.com/spyder-ide/spyder`