Further Topics in Python

by Reynaldo Morillo

Tips

- Dark blue words are links (except these words)
- While I'm presenting live, you most likely will not be able to keep up with what I'm showing. So you might be better off listening and asking questions, than trying it yourself on the spot. However, if you think you can, feel free to do so.
- This builds on the first presentation, Intro to Python
- Assumptions:
  - You have some installation of Python
  - Able to install the packages I'll use in some way
- I'll be using Anaconda

Data Science Modules

SciPy!

Essential Data Science Modules

scipy: “It provides many user-friendly and efficient numerical routines such as routines for numerical integration and optimization”.

jupyter: “The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text”.

numpy: “a library for scientific computing including a powerful N-dimensional array object, sophisticated (broadcasting) functions, tools for integrating C/C++ and Fortran code, linear algebra, Fourier transform, and random number capabilities”.

pandas: “library providing high-performance, easy-to-use data structures and data analysis tools”.

matplotlib: “Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.”

plotly: a plotting library for easily creating perhaps the best interactive graphs available today.

Lets install them!

Create Environment

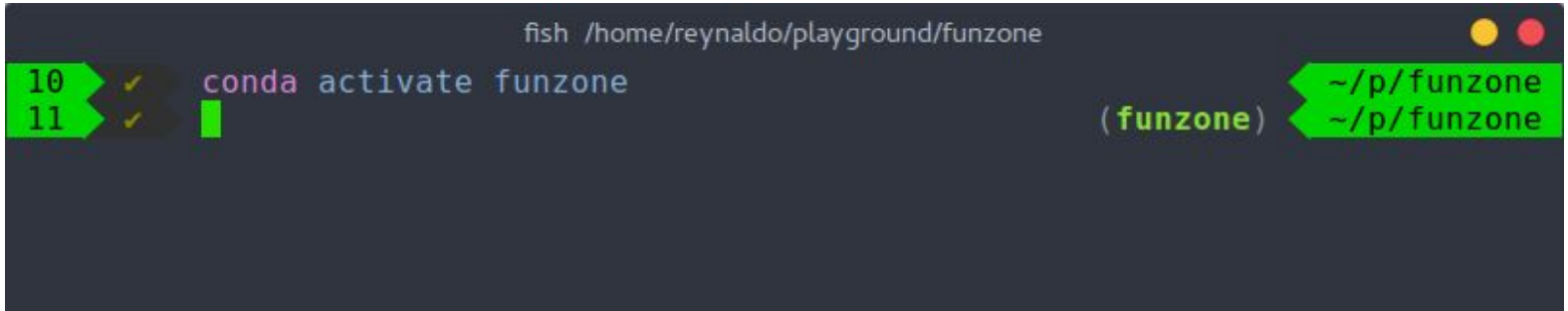
```
fish  /home/reynaldo/playground/funzone
6  ✓   conda create --name=funzone python=3.6                    ~/p/funzone
Fetching package metadata ...........
Solving package specifications: .

Package plan for installation in environment /home/reynaldo/miniconda3/envs/funz
one:

The following NEW packages will be INSTALLED:

    ca-certificates: 2017.08.26-h1d4fec5_0
    certifi:         2017.11.5-py36hf29ccca_0
    libedit:         3.1-heed3624_0
    libffi:          3.2.1-hd88cf55_4
    libgcc-ng:       7.2.0-h7cc24e2_2
    libstdcxx-ng:    7.2.0-h7a57d05_2
```

Activate Environment

```
fish  /home/reynaldo/playground/funzone
10 ✔  source activate funzone                    ~/p/funzone
```

In case you use the fish shell (like me...)

Cannot run source activate with conda in Fish-shell



Install Packages

```
fish  /home/reynaldo/playground/funzone
12 ✔  conda install scipy numpy pandas matplotlib plotly jupyter
Fetching package metadata ...........
Solving package specifications: .

Package plan for installation in environment /home/reynaldo/miniconda3/envs/funz
one:

The following NEW packages will be INSTALLED:

    asn1crypto:         0.24.0-py36_0
    bleach:             2.1.2-py36_0
    cffi:               1.11.4-py36h9745a5d_0
    chardet:            3.0.4-py36h0f667ec_1
    cryptography:       2.1.4-py36hd09be54_0
    cycler:             0.10.0-py36h93f1223_0
```

If you insist on using Windows ...

There is a way for you too.

The quickest and fastest way is using Anaconda.

This will install (practically) everything you need for data science, with Python. You just need to add them as needed to your project.

Checkout this video

SciPy

SciPy

“is a Python-based ecosystem of open-source software for mathematics, science, and engineering”

Numpy
By the Man (Travis E. Oliphant)

Pandas

Pandas

Pandas is essentially a nice wrapper on top of Numpy, that allows for powerful manipulation of the data.

It implements its own data structure that is compatible with numpy (because it uses numpy under the hood).

This is the go-to module for all your data munging (i.e. wranginling needs), which means this is a vital tool in your tool box for getting stuff done!

Series

"One-dimensional ndarray with axis labels (including time series)."

```
>>> import numpy as np
>>> import pandas as pd
>>> s = pd.Series([1,3,5,np.nan,6,8])
>>> s
0    1.0
1    3.0
2    5.0
3    NaN
4    6.0
5    8.0
dtype: float64
>>>
```

Credits to Pandas Documentation

DataFrame

"Two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). Arithmetic operations align on both row and column labels. Can be thought of as a dict-like container for Series objects. The primary pandas data structure"

```
>>> df2 = pd.DataFrame({ 'A' : 1.,
...                      'B' : pd.Timestamp('20130102'),
...                      'C' : pd.Series(1,index=list(range(4)),dtype='float32'),
...                      'D' : np.array([3] * 4,dtype='int32'),
...                      'E' : pd.Categorical(["test","train","test","train"]),
...                      'F' : 'foo' })
>>> df2
     A          B    C  D      E    F
0  1.0 2013-01-02  1.0  3   test  foo
1  1.0 2013-01-02  1.0  3  train  foo
2  1.0 2013-01-02  1.0  3   test  foo
3  1.0 2013-01-02  1.0  3  train  foo
>>>
```

Credits to Pandas Documentation

DataFrame

"Two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). Arithmetic operations align on both row and column labels. Can be thought of as a dict-like container for Series objects. The primary pandas data structure"

```
>>> df2
     A          B    C  D      E    F
0  1.0 2013-01-02  1.0  3   test  foo
1  1.0 2013-01-02  1.0  3  train  foo
2  1.0 2013-01-02  1.0  3   test  foo
3  1.0 2013-01-02  1.0  3  train  foo
>>> df2.dtypes
A           float64
B    datetime64[ns]
C           float32
D             int32
E          category
F            object
dtype: object
>>>
```

Credits to Pandas Documentation

Jupyter Notebook

Jupyter Notebook

The notebook extends the console-based approach to interactive computing in a qualitatively new direction, providing a web-based application suitable for capturing the whole computation process: developing, documenting, and executing code, as well as communicating the results. The Jupyter notebook combines two components:

A web application: a browser-based tool for interactive authoring of documents which combine explanatory text, mathematics, computations and their rich media output.

Notebook documents: a representation of all content visible in the web application, including inputs and outputs of the computations, explanatory text, mathematics, images, and rich media representations of objects.

Credits to Jupyter Documentation

Matplotlib

With hands on guide!

If that’s not enough

Here’s a link with just about everything Python and Data Science!

Fin