CHAPTER 15

PYTHON: A TOOL FOR ANALYSIS AND VISUALIZATION OF REMOTELY SENSED DATASETS

Tarun Joseph[1] and Grinson George[2]

[1]Cochin University of Science and Technology, Kochi
[2]ICAR-Central Marine Fisheries Research Institute

What is Python?

Python is a programming language which helps you to tell the computer what you want to do with the data that you have.

Why should I bother to learn Python?

- **It is extremely user-friendly**

Whether you are a beginner in using programming languages or have solid experience in programming, Python is a tool that is going to be an asset in your repository. It provides you with the basic building-blocks that you need in order to develop a solution to almost any problem that the data analysis/ visualization can throw at you.

- **A complete solution**

It has built-in packages/libraries that allow you to deal with data files with different types of files (example .csv, .txt, .xlsx, .netcdf, .grib, *etc*). So you don't have to worry about finding another software/tool in order to process a data set whose format is not very popular.

- **Automation**

If you have come across situations where you want to automate the mundane tasks such as data download, iterate a specific data analysis procedure (that you have done for one month/year) across multiple months/years, *etc.* Python can help you do so and in turn save you a lot of time that you could invest in more demanding tasks.

Ok, now where do I get Python from and how do I install it?

For Python installation there are three key aspects that a beginner should be aware of:

An Editor named Spyder

As a beginner in programming, it is recommended to have an editor that helps you to monitor the syntax and changes in variables as and when you type or run the code.

Python programming language

For you to have the full benefit of Python, it is recommended to have the Python compiler installed which will help to convert the script you write into machine-readable code which the computer can understand and execute.

Installation of additional libraries/packages

Although Python comes with a pre-installed set of libraries/packages, at times you might want to perform a particular task using Python that requires the installation of additional packages. Therefore, it is essential for the beginner to know how to install additional libraries as and when required.

All three elements come together in a distribution named Anaconda which is freely available for download and use from the following link:

https://www.anaconda.com/download/#download

The link is directed for Windows installation whose steps can be followed using this link: https://conda.io/docs/user-guide/install/windows.html. Please remember the location in which you have installed your Anaconda distribution (preferably in C drive itself)

So I got Python installed. What now? (For Windows users)

For you to start writing your first script, you need an interactive development environment (IDE) named Spyder.

To launch the Spyder editor, follow the steps mentioned below:

Step 1) Search for Terminal window by typing the keyword "cmd" in the Search window

Step 2) Select the command prompt icon (it appears as a small rectangular black icon)

Step 3) Your command prompt may look like this:

C:/Windows**>blank space**

You need to relocate the command prompt ">" to the folder where Anaconda distribution is installed. Let's say the Anaconda distribution is installed in C drive. If so, there will be a folder named Anaconda2. If that's the case type the following in the terminal window after the ">" symbol and press "Enter".

C:/Windows>cd Anaconda2

Then the command prompt will appear like this:

C:/Windows/Anaconda2>blank space

Type the following command after the ">" symbol and press "Enter"

C:/Windows/Anaconda2>cd Scripts

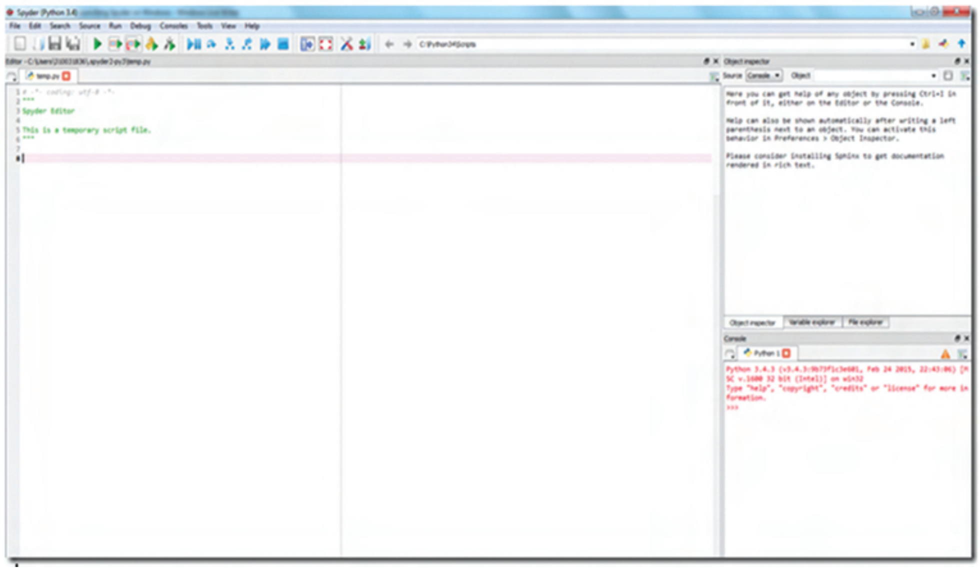
The command prompt will again change its appearance and might look like this:

C:/Windows/Anaconda2/Scripts>blank space

Type "Spyder" after the ">" symbol and press "Enter" to launch the Python editor

C:/Windows/Anaconda2/Scripts>Spyder

If you have installed your Pythonic enviroment correctly, then Spyder environment should get launched automatically and you'll see the IDE displayed on your screen (Refer Fig. 1).


Spyder (Python 3.4)
File Edit Search Source Run Debug Consoles Tools View Help
Editor - C:\Users\[illegible]\.spyder2-py3\temp.py
temp.py
Object inspector
Source Console Object
Object inspector Variable explorer File explorer
Console
Python 1

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
```

```
Here you can get help of any object by pressing Ctrl+I in
front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left
parenthesis next to an object. You can activate this
behavior in Preferences > Object Inspector.

Please consider installing Sphinx to get documentation
rendered in rich text.
```

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [M
SC v.1600 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more in
formation.
>>>
```

Now that we have Python up and running, let's begin with the programming.

For running the case studies, we require an additional libraries namely netCDF4, matplotlib and Basemap. For installation of libraries, following steps will help.

Step 1) Search for Terminal window by typing the keyword "cmd" in the Search window

Step 2) Select the command prompt icon (it appears as a small rectangular black icon)

Step 3) your command prompt may look like this:

C:/Windows**>blank space**

You need to relocate the command prompt ">" to the folder where Anaconda distribution is installed. Let's say the Anaconda distribution is installed in C drive. If so, there will be a folder named Anaconda2. If that's the case type the following in the terminal window after the ">" symbol and press "Enter".

C:/Windows>cd Anaconda2

Then the command prompt will appear like this:

C:/Windows/Anaconda2>blank space

Type the following command after the ">" symbol and press "Enter"

C:/Windows/Anaconda2>cd Scripts

The command prompt will again change its appearance and might look like this:

C:/Windows/Anaconda2/Scripts>blank space

Type "conda install netcdf4"after the ">" symbol and press "Enter" to install the netcdf4 library

C:/Windows/Anaconda2/Scripts> conda install netcdf4

The screen prompt will ask your permission to proceed once it is able to find the package required. Type "y" and press the "Enter" key.

The installation might take a while.

Similarly, you can install other libraries such as matplotlib and Basemap using the command "conda install matplotlib" and "conda install basemap" respectively.

Upon getting the required libraries installed, launch the Spyder Editor. Click on File -> Open file and find the python_data_read_visualize.py file that you have been given for this session. Click on "Open". Once it the code opens in Spyder, you can inspect it line by line and read the comments written along-side to better understand the code.

Things you should be aware of before running the script

Before you run the script, make sure you change the path of the input file in line 11 of the code and the path where the plot needs to be saved (line 57). The code is separated into smaller chunks using "#——————————————————————" symbol.

Make sure you understand each chunk individually with the help of comments, before attempting to run the code as a whole.

In order to run the statements in the code individually, select the statement and click the symbol on the top left corner of the IDE interface. Note the changes that occur simultaneously in the Ipython console. After the script gets run, if variables get created, you can monitor them in the variable explorer space.

It is a good practice to always comment using "#" symbol your code so that when you revisit your script after a few days, just by having a look at the code, you'll be able to understand the purpose of each statement in the script.

Test run

Case study 1: Read data from a NetCDF file

For this exercise, we will use a Python code named python_data_read_visualize.py. This code will read a data file and display its attributes. Additionally, it will extract the data from the file and store it in variables which can be easily visualized. Please take note of the comments which will help you to better understand what exactly the python script is doing.

Case study 2: Visualization of data

Once you know how to extract the data and store them into variables, we'll learn how to create a spatial plot and save it as per the desired resolution. The script imports the required libraries at the beginning, creates a grid using latitude and longitude values provided in the file, creates a spatial plot along with the grids (parallels and meridians), provides the title, displays the plot and saves the file in the desired format.