

# Installation Guide for PySAL and Rasterio: Hands-on Intro to Spatial Data Analysis in Python

Jenny Palomino  
FOSS4G NA 2015 Conference  
March 9, 2015

# Python Components Needed...

The compiled list below displays the minimum version required that satisfies the requirements before installing PySAL and Rasterio.

Python 2.7 is a hard requirement, as PySAL does not run in Python 3.x and Rasterio does not run in Python 2.6. In addition to Python 2.7, please install:

- numpy (1.7 or later)
- scipy (0.11 or later)
- a C compiler (more info)
- GDAL (1.9 or later)
- enum
- cligj
- affine (1.0 or later)

# Pre-packaged (and FREE) Python Distributions that satisfy some of the requirements



## WinPython for Python 2.7:

- numpy 1.9
- scipy 0.15
- PySAL: not included
- pandas 0.15
- shapely: not included
- fiona: not included
- six 1.8
- Windows only

## Anaconda for Python 2.7:

- numpy 1.9
- scipy 0.14
- PySAL 1.6
- pandas 0.14
- shapely: not included
- fiona: not included
- six 1.8
- Virtual Machine images
- Windows, Mac, Linux

## Enthought Canopy for Python 2.7:

- numpy 1.8
- scipy 0.15
- PySAL 1.7 (in academic option)
- pandas 0.15
- shapely: 1.5.1 (in academic option)
- fiona: 1.4.8 (in academic option)
- six 1.9
- Windows, Mac, Linux

# Installing a C Compiler for Rasterio

Regardless of your operating system, everyone will need a C compiler installed.

Mac and Linux operating systems often already have one installed, but it is always good to double-check. A list of free options are listed [here](#).

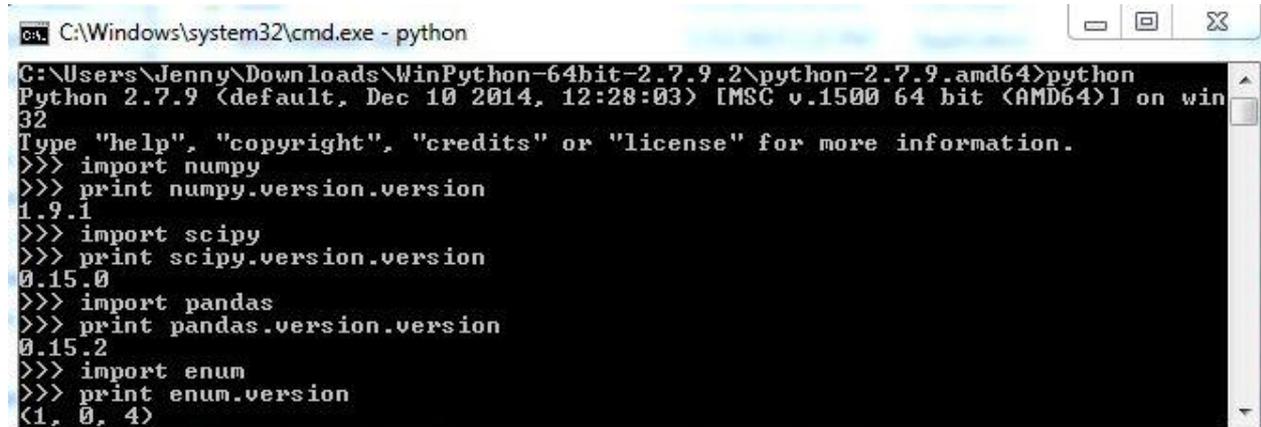
The easiest free options for Windows are [Cygwin](#) and [Microsoft Visual Studio Express for Windows Desktop](#) (you don't need to know how to use either - Rasterio, GDAL and other components will use it.)

Note: if your machine has limited space, Cygwin is the better option.

# Installation instructions for Windows

1. After installing a C compiler, begin by downloading and installing WinPython 2.7.9.1 (32 and 64 bit versions available for Python 2.7).
  - a. This install will extract a WinPython folder to wherever the installation file was downloaded
    - i. (ex: C:\Users\Name\Downloads\WinPython-64bit-2.7.9.2)
  - b. As shown in the image below, highlights of this installation include Python 2.7, numpy 1.9, scipy 0.15, pandas 0.15, enum 0.4, and six 1.8.

Note: Even if you have ArcDesktop installed, WinPython is still the easiest option. It is a self-contained Python environment that will not affect your ArcDesktop installation.



```
C:\Windows\system32\cmd.exe - python
C:\Users\Jenny\Downloads\WinPython-64bit-2.7.9.2\python-2.7.9.amd64>python
Python 2.7.9 (default, Dec 10 2014, 12:28:03) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy
>>> print numpy.version.version
1.9.1
>>> import scipy
>>> print scipy.version.version
0.15.0
>>> import pandas
>>> print pandas.version.version
0.15.2
>>> import enum
>>> print enum.version
(1, 0, 4)
```

# Windows continued...

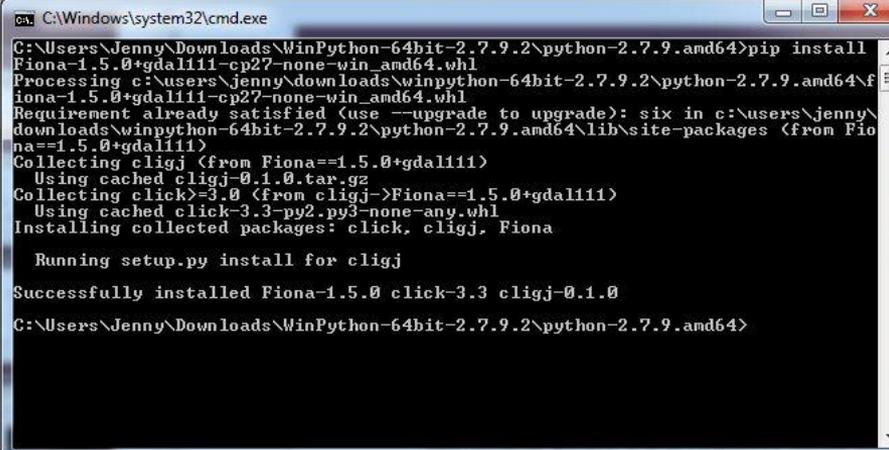
2. Next, download Fiona-1.5.0+gdal111-cp27....whl (either 32 or 64 bit for Python 2.7) and place it the WinPython folder under the subfolder called python-2.7.9.amd64.

(ex: C:\Users\Name\Downloads\WinPython-64bit-2.7.9.2\python-2.7.9.amd64)

3. Open the WinPython Command Prompt within the WinPython folder; type this command using the appropriate file name: **pip install Fiona-1.5.0+gdal111-cp27-none-win\_amd64.whl**

- a. The fiona install includes cligj.
- b. Repeat steps 2-3 for:  
GDAL-1.11.2-cp27....whl and  
rasterio-0.17.1+gdal111-cp27....whl  
(using the appropriate file names)
- b. The rasterio install includes affine.

4. Next, type this command: **pip install pysal**



```
C:\Windows\system32\cmd.exe
C:\Users\Jenny\Downloads\WinPython-64bit-2.7.9.2\python-2.7.9.amd64>pip install
Fiona-1.5.0+gdal111-cp27-none-win_amd64.whl
Processing c:\users\jenny\downloads\winpython-64bit-2.7.9.2\python-2.7.9.amd64\F
fiona-1.5.0+gdal111-cp27-none-win_amd64.whl
Requirement already satisfied (use --upgrade to upgrade): six in c:\users\jenny\
downloads\winpython-64bit-2.7.9.2\python-2.7.9.amd64\lib\site-packages (from Fio
na==1.5.0+gdal111)
Collecting cligj (from Fiona==1.5.0+gdal111)
  Using cached cligj-0.1.0.tar.gz
Collecting click=3.0 (from cligj->Fiona==1.5.0+gdal111)
  Using cached click-3.3-py2.py3-none-any.whl
Installing collected packages: click, cligj, Fiona
  Running setup.py install for cligj
Successfully installed Fiona-1.5.0 click-3.3 cligj-0.1.0
C:\Users\Jenny\Downloads\WinPython-64bit-2.7.9.2\python-2.7.9.amd64>
```

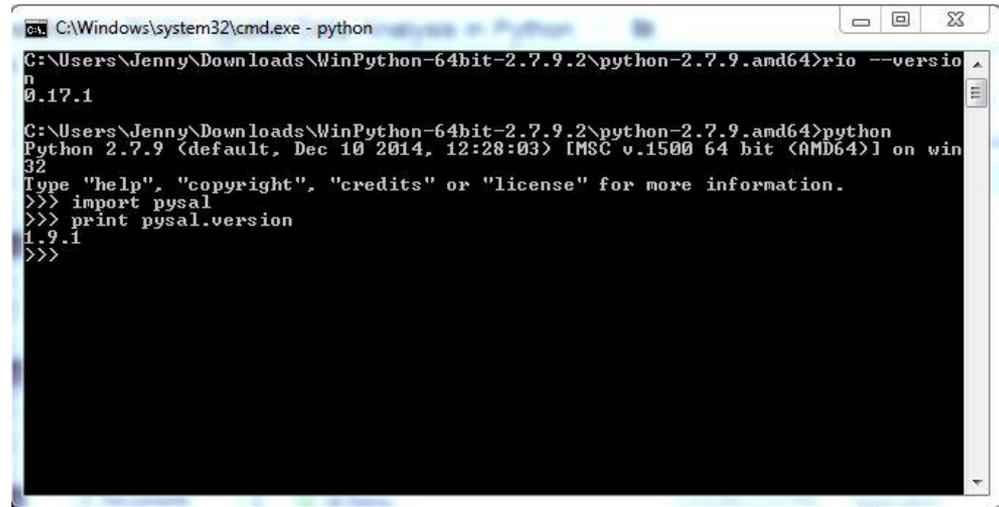
# Windows: Test PySAL and Rasterio

To test success of installation, type these commands into the WinPython Command Prompt:

1. `rio --version`
  - a. it will read 0.17.1

Then, to test PySAL, type:

1. `python`
2. `import pysal`
3. `print pysal.version`
  - a. it will read 1.9.1



```
C:\Windows\system32\cmd.exe - python
C:\Users\Jenny\Downloads\WinPython-64bit-2.7.9.2\python-2.7.9.amd64>rio --version
0.17.1
C:\Users\Jenny\Downloads\WinPython-64bit-2.7.9.2\python-2.7.9.amd64>python
Python 2.7.9 (default, Dec 10 2014, 12:28:03) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import pysal
>>> print pysal.version
1.9.1
>>>
```

# Installation instructions for Mac and Linux

The best option for users without their Python 2.7 environment set-up is to install Enthought Canopy. In addition to the free version, there is a full installation available at no cost to academics. For Mac or Linux users who already have their Python 2.7 environment set-up, be sure to review instructions are available on the installation pages for PySAL and Rasterio.

Highlights of Canopy include:

- Python 2.7
- numpy 1.8
- scipy 0.15
- PySAL 1.7 (included in Academic option)
- pandas 0.15
- shapely: 1.5.1 (included in Academic option)



After installing Canopy, install PySAL. Then, install GDAL (1.9 or later), enum, cligj, and affine (1.0 or later) before installing Rasterio. All of these Python components can easily be installed via the terminal by typing: **pip install pysal**, **pip install GDAL**, **pip install enum**, etc

# Mac and Linux: Test PySAL and Rasterio

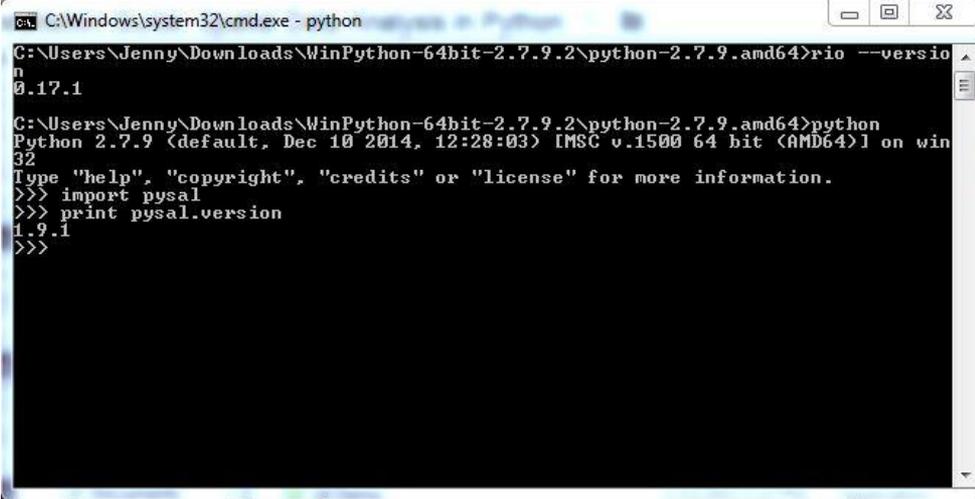
Note: you can test the installs via the terminal, but please make sure that you have a text editor or IDE that can run Python, like IDLE, Sublime, PyCharm, etc. (Canopy includes its own IDE)

To test success of installation, type these commands into the terminal:

1. `rio --version`
  - a. it will read 0.17.1

Then, to test PySAL, type:

1. `python`
2. `import pysal`
3. `print pysal.version`
  - a. it will read 1.7 if using Canopy
  - b. it will read 1.9 if you installed from pip



```
C:\Windows\system32\cmd.exe - python
C:\Users\Jenny\Downloads\WinPython-64bit-2.7.9.2\python-2.7.9.amd64>rio --version
0.17.1
C:\Users\Jenny\Downloads\WinPython-64bit-2.7.9.2\python-2.7.9.amd64>python
Python 2.7.9 (default, Dec 10 2014, 12:28:03) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import pysal
>>> print pysal.version
1.9.1
>>>
```

# Trouble-shooting PySAL and Rasterio Installs

If you get any error messages during testing, check your dependencies (sometimes the messages will clearly tell you that something is missing).

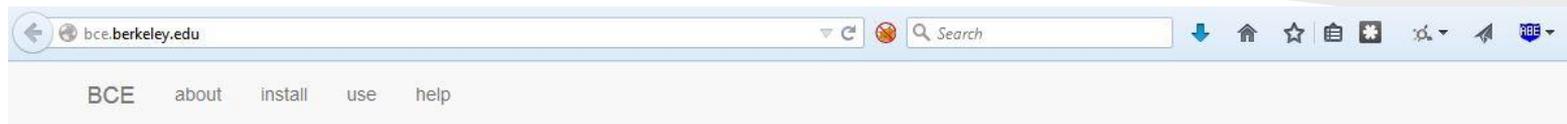
For Rasterio: the issue is often due to a missing C compiler or missing GDAL.

For PySAL: the issue is often due to missing numpy (especially if you did not use a pre-packaged Python distribution like Canopy, Anaconda or WinPython).

If the command **pip install** results in a message that pip cannot be found, you can install pip (more info on using pip).

Last, another common issue is when multiple versions of Python are installed on one machine. When testing, make sure to use your Python 2.7 environment where these items are installed.

# Berkeley Common Environment - Virtual Machine for Scientific Computing



## What is BCE?

BCE stands for the Berkeley Common (or Compute, or Collaborative...) Environment. It is designed to provide a common Linux computational environment for classwork and research.

## BCE Vision

The goal for the BCE is to provide both the ready-made environments and also the “recipes” or scripts setting up these environments. It should be easy for a competent Linux user to create recipes for custom tools that might not be broadly useful (and thus, not already in BCE).

BCE is designed for classwork and research in the sciences at Berkeley, broadly defined to include social science, life science, physical science, and engineering. Using these tools, users can start up a virtual machine (VM) with a standardized Linux operating environment containing a set of standard software for scientific computing. The user can start the VM on their laptop, on a university server, or in the cloud. Furthermore, users will be able to modify the instructions for producing or modifying the virtual machine in a reproducible way for communication with and distribution to others.

We envision the following core use cases:

- creating a common computing environment for a course or workshop,
- creating a common computational environment to be shared by a group of researchers or students, and
- disseminating the computational environment so outsiders can reproduce the results of a group.

## Getting help with BCE

- **Installation:** Instructions for self-paced installation.
- **Next BCE install sessions:** During scheduled sessions, experts are available to help you install and get familiar with BCE. The next session is: **None Scheduled at this time; please request help via the [support forum](#)**
- **Support Forum:** Ask the community for support via email.

# OSGeo Live - Virtual Machine for Open Source and Web GIS



# List of Useful Python Spatial Packages

## *Data Handling:*

[shapely](#)

[GDAL/OGR](#)

[pyQGIS](#)

[pyshp](#)

[pyproj](#)

## *Analysis:*

[shapely](#)

[numpy](#), [scipy](#)

[pandas](#), [GeoPandas](#)

[PySAL](#)

[Rasterio](#)

[scikit-learn](#), [scikit-image](#)

## *Plotting:*

[matplotlib](#), [prettyplotlib](#)

[descartes](#)

[cartopy](#)

[https://github.com/SpatialPython/spatial\\_python/blob/master/packages.md](https://github.com/SpatialPython/spatial_python/blob/master/packages.md)  
<http://spatialdemography.org/essential-python-geospatial-libraries/>  
<http://carsonfarmer.com/2013/07/essential-python-geo-libraries/>

# Online (and free!) Resources for Python

*Code Academy (programming tutorials):* <http://www.codecademy.com/>

*Coursera (full courses):* <https://www.coursera.org/courses?query=python>

*Python wiki pages:* <https://wiki.python.org/moin/BeginnersGuide/NonProgrammers>  
<https://docs.python.org/2/tutorial/>

*Python at Berkeley (DLab):* [http://python.berkeley.edu/learning\\_resources.html](http://python.berkeley.edu/learning_resources.html)

*Python Books and Training:* <http://pythonbooks.revolunet.com/>  
<http://www.learnpython.org/>

*ArcPy tutorials from ESRI:* <http://training.esri.com/gateway/index.cfm?fa=catalog.webCourseDetail&courseid=2520>

<http://training.esri.com/gateway/index.cfm?fa=catalog.webCourseDetail&courseid=2523>

# Questions or Issues with Installation?

Email me at [jpalomino@berkeley.edu](mailto:jpalomino@berkeley.edu)