Week 13 Lecture:
Python Package Management and Extension

Introduction to Programming for GIS & Remote Sensing
GEO6938-1469
GEO4938-147A
Re-cap From Past Lectures

• Last week:
  – Picked up on how to manipulate different aspects of ArcGIS documents and data
    • Geometry (tokens for update cursors, etc.)
    • Map layers, symbology and more
This Week: Building on Python’s Functionality

• Programming languages are as good as they are extensible
  – Most “general purpose” programming/scripting environments have package repository and management system:
    • Perl: CPAN
    • Ruby: Gems
    • Java: Maven
    • TeX: CTAN
    • R: CRAN
    • Python: Python Package Index (PyPI) - pip & eggs
Python Package Index

Dandy... but how do we interface with it?
Used to be “eggs” but now: pip

• If you run Linux or Mac OSX this is easy:
  – Install or upgrade “setuptools” and then...
  – python get-pip.py
  – Follow instructions here:

• If you run Windows, the above can be tough
  – Luckily, we have a very simple tool: pip-Win
    • https://sites.google.com/site/pydatalog/python/pip-for-windows
pip Example – Network Analysis

• Search PyPI for “network graph”
Follow along... Example 1

• We will install the Python networkx package from PyPI using pip-Win

• Check out the /src folder for example code
Many Packages Written in Python

• And these packages can be added without problem as long as dependencies are met
  – Dependencies can be an issue with ArcGIS’ Python
    • numpy and matplotlib installed by default, versions can be crusty...

• However... many packages rely on native C/C++ or some other compiled code to run...
  – We (or someone else) needs a compiler!
Compiled Code

• Again, Linux and Mac OSX people have it relatively easy
  – Linux has gcc and Mac has XCode (also gcc)
  – pip handles compilation for installing packages from source relatively painlessly

• On Windows... tricky business...
  – Visual Studio 2008, 2010-2013... all provide VC compilers
    • However, consistently can cause problems when compiling across different versions and with alt. libraries
Installing Packages From Source: Windows

• Visual C++ Studio is a free download from MSDN, but it can be tricky to configure
  – PATH and environment settings
On Windows...

• ... much easier to deal with pre-compiled binary packages
  – Package websites often offer binaries
    • e.g. matplotlib, numpy, scipy
  – Or rely on someone else to compile them for you:
    • ... do you trust them?

• The major downside:
  – Must match your Python version and the versions of all package dependencies
Follow along... Example 2

- We will install the scipy package from a binary distribution

- Check out the /src folder for example code

- scipy provides a lot of added scientific computing functionality:
Some “Packages” Very Platform Specific

• ... and are not available for installation through PyPI
• You must compile from source or install a binary package
• Many environments for scientific computing (e.g. Sage, Python(x,y), Enthought, etc.) not only group packages, but also an entire Python environment:
  – https://wiki.python.org/moin/NumericAndScientific
Follow along... Example 3

• We will install the PyGame binary group of packages

• Check out the /src folder for example code

• PyGame provides a lot of added functionality for advanced visualization, simulation, user interactions and, well... game building:
Follow along... Examples 4 and 5

- We will install the PypeR package using pip
- Check out the /src folder for example code
- PypeR is one of several packages that allows us to interface Python with R:
  - [http://www.webarray.org/softwares/PypeR/](http://www.webarray.org/softwares/PypeR/)
Next Week

• We will spend more time talking about spatial data and Python
  – Specifically how to leverage R and its spatial data analysis with Python
  – How to get stuff done without ArcGIS
  – More details on Python packages of note