Jeffrey Tharsen tharsen@uchicago.edu
"Spatial Data Analysis" RCC Workshop
Fall 2018

Sources: https://github.com/rcc-uchicago/env_bootcamp/

I. To start Jupyter Notebook on Midway (Linux only) :

a. Get the .zip file from Sources above;
   Move the file run_ipython.sh (in Day7/pythonforspatialanalysis_rcc_materials) to Midway.
b. ssh in to Midway (ssh username@midway2.rcc.uchicago.edu)
c. run "sh run_ipython.sh" to start Jupyter Notebook Server
d. copy URL, paste into your local browser

****** ArcGis (including python libraries arcgis and arcpy) cannot be run from Midway ******

II. To set up Anaconda3 and the Jupyter Notebook on your laptop (Windows and Mac OS) :

a. In a browser, go here, download and install the newest version of Anaconda3 for your OS:
   https://www.anaconda.com/download/

b. Then from a command line (Terminal in Mac, or Command Prompt in Windows):

```
pip install arcgis
pip install pysal
jupyter notebook
```

III. RCC GIS Website (ArcGIS, ESRI geocoding service, lots of resources and links) :

https://gis.rcc.uchicago.edu/

IV. ArcGIS Online UChicago Portal :

https://uchicago.maps.arcgis.com/home/signin.html

V. Using the Google Geocoding Service to geocode locations (addresses) :

In Jupyter Notebook, open: pythonforspatialanalysis_rcc_materials/google_geocoding/**googlegeocoding.ipynb**
Or, in python, run **googlegeocoding.py**

The output file **villages_1815_output.csv** can then be imported into **ArcGIS Online** (or any mapping system).

VI. Calculating distances using python.math:

In Jupyter Notebook, open: pythonforspatialanalysis_rcc_materials/ measuredistances/**measuredistances.ipynb**
Or, in python, run **measuredistance1.py, measuredistance2.py, measuredistance3.py**

VII. Main toolkits for spatial data analysis with python:

a. **ArcGIS arcgis** -- API for ArcGIS functions (python library, sample notebooks)
   https://developers.arcgis.com/python/guide/

"The ArcGIS API for Python is a powerful, modern and easy to use Pythonic library to perform GIS visualization and analysis, spatial data management and GIS system administration tasks that can run both in an interactive fashion, as well as using scripts.

It enables power users, system administrators and developers to leverage the rich SciPy ecosystem for automating their workflows and performing repetitive tasks using scripts. It integrates well with the Jupyter Notebook and enables academics, data scientists, GIS analysts and visualization enthusiasts to share geo-enriched literate programs and reproducible research with others.

This guide describes how to use the ArcGIS API for Python to write Python scripts, incorporating capabilities such as mapping, query, analysis, geocoding, routing, portal administration, and more. A great place to start developing once you've installed the API is to browse the sample notebooks."

(A) Install/load Anaconda3
(B) Install arcgis package: "pip install arcgis" or "conda install -c esri arcgis"
(C) Test in Jupyter Notebook : pythonforspatialanalysis_rcc_materials/arcgis/arcgis_test.ipynb

```
from arcgis.gis import GIS
my_gis = GIS()
my_gis.map()
```

[screenshot **arcgis example screenshot 1.png** = success]

(D) Test Example: https://github.com/Esri/arcgis-python-api/blob/master/samples/04_gis_analysts_data_scientists/analyze_new_york_city_taxi_data.ipynb

[screenshot **arcgis example screenshot 2.png** = success]

(E) Sample Notebooks for Spatial Data Analysts: https://github.com/Esri/arcgis-python-api/tree/master/samples/04_gis_analysts_data_scientists

Main page: https://developers.arcgis.com/python/sample-notebooks/

(F) API Reference Guide: https://esri.github.io/arcgis-python-api/apidoc/html/

b. ArcGIS ArcPy -- scripting ArcGIS (from within ArcGIS)
http://pro.arcgis.com/en/pro-app/arcpy/get-started/what-is-arcpy-.htm

"ArcPy is a Python site package that provides a useful and productive way to perform geographic data analysis, data conversion, data management, and map automation with Python.

This package provides a rich and native Python experience offering code completion (type a keyword and a dot to get a pop-up list of properties and methods supported by that keyword; select one to insert it) and reference documentation for each function, module, and class.

The additional power of using ArcPy is that Python is a general-purpose programming language. It is interpreted and dynamically typed and is suited for interactive work and quick prototyping of one-off programs known as scripts while being powerful enough to write large applications in. ArcGIS applications written with ArcPy benefit from the development of additional modules in numerous niches of Python by GIS professionals and programmers from many different disciplines."

c. GeoDa/Luc Anselin PySAL -- calculation of "spatial weights" from geomatrices (python library)
https://pysal.readthedocs.io/

(A) Install pysal first: pip install pysal (or "pip install --user pysal" on Midway)

(B) In Jupyter Notebook, run: pythonforspatialanalysis_rcc_materials/pysal/moran.ipynb

or from any command line:

$ python moran.py (requires sourcefiles **stl.txt**, **stl.gal** be in same directory)

```
w = <pysal.weights.weights.W object at 0x1123cf358>
0.244
-0.012987012987012988
0.00014
```

"It is important to underscore what PySAL is, and is not, designed to do. First and foremost, PySAL is a library in the fullest sense of the word. Developers looking for a suite of spatial analytical methods that they can incorporate into application development should feel at home using PySAL. Spatial analysts who may be carrying out research projects requiring customized scripting, extensive simulation analysis, or those seeking to advance the state of the art in spatial analysis should also find PySAL to be a useful foundation for their work.

End users looking for a user friendly graphical user interface for spatial analysis should not turn to PySAL directly. Instead, we would direct them to projects like STARS and the GeoDaX suite of software products which wrap PySAL functionality in GUIs. At the same time, we expect that with developments such as the Python based plug-in architectures for QGIS, GRASS, and the toolbox extensions for ArcGIS, that end user access to PySAL functionality will be widening in the near future."