Introduction to GIS Methods in Economics

Giorgio Chiovelli Sebastian Hohmann

Bonn, 27/05/2019

Table of contents

Overview

The plan for today

Introduction to Python

- Installing python
- Language peculiarities
- Opening and closing python
- Interactive mode and normal mode

Turning a Model-Builder "model" into python code

- U.S. counties average agricultural suitability example

Python

A sales pitch

Even if you don't end up using GIS, python may be for you

- Fully fledged programming language, very easy to learn
- Jupyter for seamlessly combining beautifully formatted text and code: reproducible research! (can do symbolic math, so even for theorists!)

  `http://jupyter.org/`

  - Paul Romer (heard of him?) is a fan:
    `https://paulromer.net/jupyter-mathematica-and-the-future-of-the-research-paper/#more-3616`
- The "scientific stack": *numpy, pandas, scipy, matplotlib, Scikit-learn,...*
- Very promising (still in alpha, Mac users will have easier time trying it out than Win users:) *pydatatable*
- `https://www.youtube.com/watch_popup?v=1yTHSxJ4KL8`
- `https://github.com/h2oai/datatable`

Introduction to Python

Installing python

ArcGIS comes with python

- If you have installed ArcGIS, you already have python.

The official source

- `https://www.python.org/downloads/`

A nice distribution

- Anaconda
- `https://www.continuum.io/downloads`
- Has most packages you will need if you want to use python for research.

2.x.y or 3.x.y ?

- The two versions are maintained and widely used in parallel.
- For the basic functionalities we will use, the differences are minimal.
- ArcGIS works in python 2.7.x so that is what we will use.

Introduction to Python

Language peculiarities

Case sensitive

- `a = 2` is different from `A = 2`

Indentation is syntactically significant

- You don't need to enclose blocks in { } as in Stata, or terminate blocks with a statement like `end` in MATLAB.
- Indent to start a block, dedent to end it.
- Statements that should be followed by indentation start with a colon ":".

Path names

- Use frontslashes or double backslashes or raw strings (a backslash inside a raw string is just a backslash, otherwise backslash is used to escape special characters).
- `'C:/the/path/to/your/folder'`
- `'C:\\the\\path\\to\\your\\folder'`
- `r'C:\the\path\to\your\folder'`

Windows

- Start → Search → "cmd" → Enter
- Type `python` and hit Enter

Mac

- Search for "Terminal" and open it
- Type `python` and hit Enter

Closing python

- Type `quit()`

We will now switch over to python. The examples we will run are in the file *python_intro.py* on google drive.

Introduction to Python

Interactive mode and normal mode

Interactive mode

- Open python
- You see a prompt >>>
- Type the examples in section 1) - 6) from *python_intro.py* into the command line / terminal.
- In interactive mode, you get immediate feedback for each statement. Previously run statements (such as variable assignments) are kept in memory.

Normal mode

- Write a python script and save it with the ending *.py* in some directory.
- Inside the command line / terminal, type `cd` / `pwd` to see your current working directory.
- Change to the directory containing the *.py* file using `cd path/to/directory`.
- Run the script by typing `python scriptname.py`
- All the commands in the script are executed, just like in a *.do* or *.m* file.
- Alternatively, you can use **IDLE** (in the Start menu under ArcGIS → Python 2.7 → IDLE (Python GUI)).

Agricultural suitability in U.S. counties example

Saving Model Builder model to python script

Exporting to Python

- Inside the ArcGIS Model Builder, click Model → Export → To Python script...
- Save it with some name ending in *.py*.
- This will give you a raw python script, which you can edit further to make it more readable and make it run smoothly.

Try to run the script

- Inside the command prompt, typing `python suit_raw.py` produces an error.
- Let's open the script to fix it.

Agricultural suitability in U.S. counties example

Cleaning up the script, part 1

Open the script in an editor

- in IDLE, click File → Open...

The header

- Delete the comments at the top (before the `#Import arcpy module`)
- Before getting to the definition of local variables, enter (adjust the path names to fit your directory structure):

```
maindir = 'D:/Dropbox/PoliteconGIS/Zurich/Lecture 2'
GISdir = 'maindir + /GISdata'
outdir = 'maindir + /Lecture 2 part 2/_output'

arcpy.env.workspace = outdir
arcpy.env.overwriteOutput = True
arcpy.CheckOutExtension("Spatial")
```

Agricultural suitability in U.S. counties example

Cleaning up the script, part 2

Local variables

- Local variables are assignments to variables that we later put into geoprocessing functions as arguments.
- Mostly, these are just path names pointing to shapefiles etc.
- We could directly type these path names as arguments into the functions, but this would make the code unreadable.
- A lot of variables created by the auto-generated code are duplications (result of input and output in the Model Builder being the same, e.g. for *Define Projection*).
- $\Rightarrow$ We can reduce these duplications.
- Similarly, all the variables have full path names.
- $\Rightarrow$ We can shorten these.
- For non-input files that will be generated by geoprocessing, we don't have to give a pathname. ArcGIS will store them in the workspace defined above.
- For input files, combine the `GISdir` defined in the preamble with the file name (recall how to concatenate strings).

Agricultural suitability in U.S. counties example

Cleaning up the script, part 3

Geoprocessing

- It is a good idea to comment out all geoprocessing commands in the beginning, run the script; then (if it works), successively un-comment one geo-processing command after another.
- As geoprocessing scripts can run for a while, it is a good idea to insert messages using `print 'xyz'`.
- Some geoprocessing tools are very long in python. We can usually shorten these to make the code more readable. E.g. for *Define Projection* by first defining the spatial reference as a new variable and then inserting that variable instead of the long string definition*.
- The syntax of the geoprocessing methods is an intuitive translation of the menu-structure of ArcGIS into python scripting language. Essentially, every field you fill out in the ArcGIS GUI corresponds to an argument of the corresponding geoprocessing method.
- To find out how the syntax of a given tool works, consult the online help. E.g. for *Select*:
  `http://desktop.arcgis.com/en/arcmap/10.5/tools/analysis-toolbox/select.htm`

*For a list of coordinate systems, see:

`http://resources.arcgis.com/en/help/arcgis-rest-api/index.html#/Using_spatial_references/02r3000000qq000000/`

Agricultural suitability in U.S. counties example

Cleaning up the script, part 4

Zonal Statistics problem

- We have encountered a conflict between *Zonal Statistics* and declaring the workspace environment. Sometimes the only way around this seems to be to separate the python scripts into a part doing zonal statistics and a part doing everything else.
- Try to run the two parts in one script (as we have given it to you). The crashes don't happen on all computers.

Exiting ArcGIS

- After the last geoprocessing command, add
  `arcpy.CheckInExtension('Spatial')`
  `del arcpy`

Cleanup

- since we only care about the final `.csv` output, we can clean up the intermediate files as follows

```
import os
for fn in os.listdir(outdir):
    if '.csv' not in fn:
        try:
            os.remove(os.path.join(outdir,fn))
        except:
            continue
del os
```