CODERS' CORNER

csae
CENTRE FOR THE STUDY OF AFRICAN ECONOMIES

UNIVERSITY OF OXFORD

COMPUTING SPATIAL STATISTICS IN ARCGIS

ArcGIS is a powerful (albeit oft-temperamental) piece of software, which can be used to do any and all spatial data manipulation. For example, you can use ArcGIS to find the shortest distance between any two cities embedded in a network, to calculate the number of hospitals within a radius around a location, or to join data sets at different underlying geographies. But, to utilize its full power and produce reproducible work, it's necessary to code in it. You can write scripts to run using ArcGIS with python and ArcGIS's "arcpy" package. To do this you will first need to download python, I highly recommend using anaconda for this, and an IDE for which I recommend pycharm.

I'll use an example to walk through the setup and illustrate how simple spatial statistics can be computed using ArcGIS and python. In this example I'm interested in calculating average potential crop yields in each Commune[1] of Benin over time.

To do this I've downloaded a .csv file from GAEZ which consists of meta data (i.e. year crop type) and a link to the .tif file of yields in raster format. Raster basically means an image, in this case an image with roughly 9x9km pixels[2]. My problem is that I'm interested in yields at a coarser geography[3], and only in Benin. Thus, the plan is to loop over each data-set, calculate the average yield within each Commune and export a spreadsheet of this information for each year and crop. In short, I want to convert the upper excel file depicted below which contains links to each .tif file into a series of excel files that look like the lower one which contains information on mean yields by crop by commune (that is by geolevel2).

	A	F	I	K	L	N	O
1	Name	Time Period	Crop	Input Level	Data Units	Download URL	File Identifier
2	alfa200b_yld	1981-2010	Alfalfa	Low	10kg DW/ha	https://s3.eu-west-1.amazona	277243
3	bana200b_yld	1981-2010	Banana	Low	kg DW/ha	https://s3.eu-west-1.amazona	280422
4	barl200b_yld	1981-2010	Barley	Low	kg DW/ha	https://s3.eu-west-1.amazona	282691
5	bckw200b_yld	1981-2010	Buckwheat	Low	kg DW/ha	https://s3.eu-west-1.amazona	282713
6	bean200b_yld	1981-2010	Phaseolus bean	Low	kg DW/ha	https://s3.eu-west-1.amazona	282719
7	bhsg200b_yld	1981-2010	Biomass highland sorghum	Low	kg DW/ha	https://s3.eu-west-1.amazona	281736
8	blsg200b_yld	1981-2010	Biomass lowland sorghum	Low	kg DW/ha	https://s3.eu-west-1.amazona	278110

	B	C	D	E	F
1	GEOLEVEL2	ZONE_CODE	COUNT	AREA	MEAN
2	204001001	1	54	0.375	1260.037037
3	204001002	2	59	0.409722222	1245.576271
4	204001003	3	43	0.298611111	1243.906977
5	204001004	4	69	0.479166667	720.8985507
6	204001005	5	40	0.277777778	1046.425
7	204001006	6	55	0.381944444	1291.381818
8	204002001	7	12	0.083333333	1373.833333
9	204002003	8	47	0.326388889	1309.723404
10	204002002	9	10	0.069444444	1340.6
11	204002004	10	36	0.25	1391.722222
12	204002005	11	21	0.145833333	1317.142857
13	204002006	12	17	0.118055556	1435.647059
14	204002007	13	25	0.173611111	1366.08

[1]In Benin, Communes are the second administrative division.
[2]Pixels are 5 arc-minute (about 9 x 9 km at the equator) grid cells.
[3]That is, I'm interested in calculating yields at the commune level which comprises of several (many) pixels.

First, we import packages including arcpy and some other python packages that are needed for the example. Then we set some parameters, including the parallelisation, over-write (whether we overwrite existing files or not) and the home directory as well as “checking out” the necessary extensions in ArcGIS.

```python
# Import packages
import os, arcpy
from arcpy import env
from arcpy.sa import *
import sys
import time
import urllib
from csv import DictReader
```

```python
# Set parallelisation parameter as a percentage of available cores
arcpy.env.parrallelProcessingFactor = '100%'

# Check out needed extensions
arcpy.CheckOutExtension("Spatial")

# Allow over-write
arcpy.env.overwriteOutput = True

# Set home directory
dir = os.path.join('E:/', 'Dropbox', 'home_dir')
arcpy.env.workspace = dir

```

Having dealt with the housekeeping above we turn to the code proper which is given below. The code is annotated to explain what each line is doing. We loop over each row in the excel file, extract meta information, and then download the associated dataset using the link provided. Once downloaded we perform the spatial operation “ZonalStatistic-sAsTable” which calculates the mean yield across pixels in each of the larger communes in Benin. This command can generate a wide variety of statistics other than the mean. For example, if we are interested in the maximum value, we can change ’MEAN’ in line 55 of the code below to say ’MAX’. Finally, I export the extracted information to a .csv file.

```python
# steps:(1) loop over every link (dataset)
#       (2) download tif file
#       (3) calculate average pixel in each geolevel2
#       (4) export resulting data to csv (geolevel2 - year - crop - high/low - yield)

# read csv file downloaded from GAEZ and loop through rows
ii = 0
with open(os.path.join('E:/', 'Dropbox', 'home_dir', 'links_to_data.csv'), 'r') as read_obj:  # open the .csv file
    datasets = DictReader(read_obj)  # maps the .csv file contents into a python dictionary
    for row in datasets:  # loop over each row in the dataset

        # keep track of where we are in the loop
        ii = ii + 1
        print('{num}'.format(num=ii))

        # get meta data using python dictionary loop ups
        year = row['Time Period']
        crop = row['Crop']
        input_level = row['Input Level']

        # get download URL, this is where the yield raster file can be found
        url = row['Download URL']

        # save .tif file to disk
        file_name = os.path.join('E:/', 'Dropbox', 'home_dir',
                                 'yield_{crop}_{input}_{year}.tif'.format(crop=crop, input=input_level, year=year))
        testfile = urllib.URLopener()  # open the url
        testfile.retrieve(url, file_name)  # retrieve the contents of the url and save in file name

        # calculate average pixel in each area, use a .shp file (shape file) of locality boundries
        g2 = os.path.join('E:/', 'Dropbox', 'home_dir', 'shape_boundries.shp')  # location of shape file
        table = os.path.join('E:/', 'Dropbox', 'home_dir', 'table')  # set where to save the table
        arcpy.gp.ZonalStatisticsAsTable_sa(g2, 'GEOLEVEL2', file_name, table, 'DATA',
                                           'MEAN')  # perform the spatial operation
        # g2 = location of shape file, GEOLEVEL2 = name of ID variable in shape file for each shape
        # file_name = location of .tif file on disk, table = where to save the table
        # DATA is an option that indicates we should ignore missing data values (not a problem in this setting)
        # MEAN indicates that I want the mean values to be calculated

        # export to csv
        csv_location = os.path.join('E:/', 'Dropbox', 'home_dir', 'crop_intermediate')  # location of .csv export file
        csv_name = 'yield_{crop}_{input}_{year}.csv'.format(crop=crop, input=input_level, year=year)  # name of file
        arcpy.TableToTable_conversion(table, csv_location, csv_name)  # arc operation to extract data to .csv
```

Luke Milsom, DPhil candidate in the Department of Economics, University of Oxford
08 December 2021