Python script for dynamic threshold identification from NDVI by county FIPS in the CONUS

Updated on: Sep. 9, 2014

# Introduction

The python scripts have the purpose to identify a NDVI threshold value by county in the CONUS. These scripts require as user input arguments county FIPS and NDVI raster datasets, an output directory path that will be used as workspace and to save the final output, FIPS and USDA estimated irrigation area in acres field names from the county FIPS dataset. The first python script (DynamicThresholdIdentification.py) provides an ArcGrid as output with just two fields, while the second python script (DynamicThresholdIdentification\_Validation.py) provides an IMG file as output with fields with the areas in acres for each NDVI value for validation purposes.

## Requirements

ArcGIS 10.1 or later

Python 32-bit version

Python modules: *dbfpy* (python module for python 32-bit version)

ArcGIS 10.1 uses the 32-bit version (if the 64-bit version has not been installed), while ArcGIS 10.2 uses the 64-bit version. However, the python 32-bit is installed and can be used in ArcGIS 10.2 (see examples below).

### User Inputs

* Peak\_NDVI: raster dataset with integer NDVI values
* CountyFIPS: raster dataset with county fips pixels and attribute table that contains information with the FIPS codes for each county and the estimated USDA irrigation areas in acres.
* Output\_Dir: output directory where the results and intermediate data can be found/
* CountyFIPS\_fieldname: field name from the county FIPS raster dataset that contains the FIPS code information for each county
* USDA\_Areas\_fieldname: field name from the county FIPS raster dataset that contains the USDA estimated irrigation area information in acres.

### NDVI raster dataset requirements

Coordinate System = Sphere\_ARC\_INFO\_Lambert\_Azimuthal\_Equal\_Area

Extent = -2050500 -2136500 2536500 752500 NaN NaN NaN NaN (CONUS)

Cell Size = 250 (meters)

# Python scripts

## 1) DynamicThresholdIdentification.py

The output of this script is an updated county FIPS ArcGrid with two fields, "NDVIThrAbv" and "PeakNDVIAc", where "PeakNDVIAc" contains the cumulative area reached by the NDVI threshold value, and "NDVIThrAbrv" contains the NDVI threshold value. If the cumulative area never reaches the estimated USDA irrigation area, the NDVI threshold value will be the minimum NDVI value. Moreover, if the estimated USDA irrigation area is equal to 0, the NoData = 999 values is assigned to the NDVI threshold value. The NDVI threshold value is integer, while the PeakNDVI values is a float with 2 decimal digits.

Usage:

<python> DynamicThresholdIdentification.py [Peak\_NDVI] [CountyFIPS] [Output\_Dir] [CountyFIPS\_fieldname] [USDA\_Area\_fieldname]

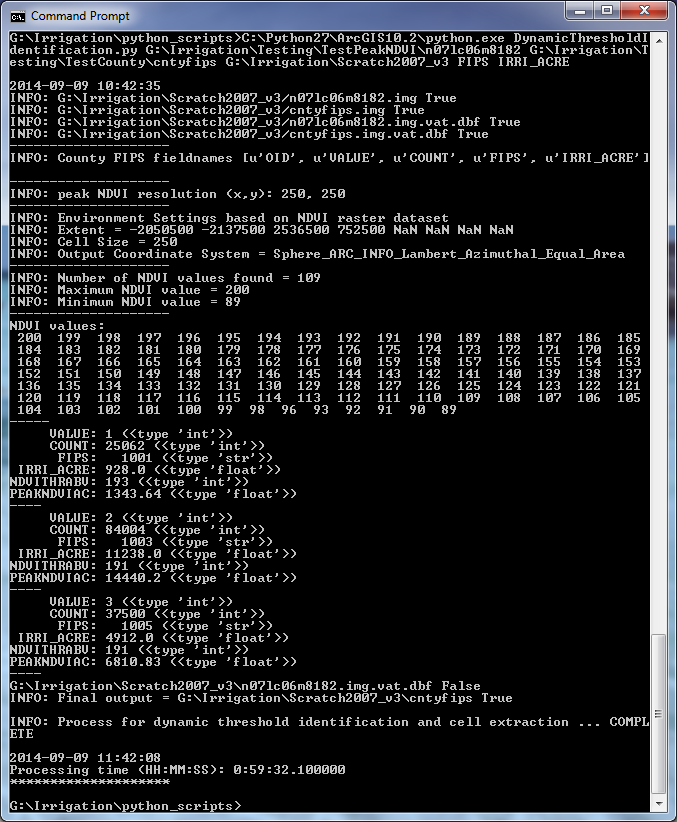
## 2) DynamicThresholdIdentification\_Validation.py (for validation purpose)

This python scripts runs and requires the same inputs as the "DynamicThresholdIdentification.py" script, but the output is different. For this script, the output is an updated county FIPS raster dataset as an IMG file with the fields "NDVIThrAbv", “PeakNDVIAc", and a fields for the area in acres for each NDVI value in the associated attributed table. The table can be useful for validation/analysis purposes and it can be open in Excel.

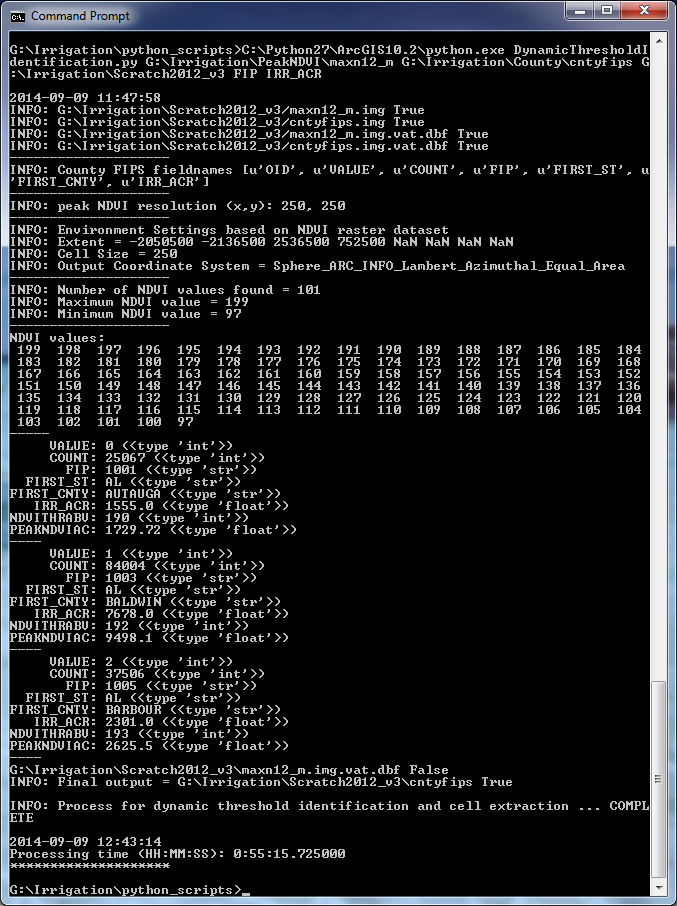
Usage:

<python> DynamicThresholdIdentification\_Validation.py [Peak\_NDVI] [CountyFIPS] [Output\_Dir] [CountyFIPS\_fieldname] [USDA\_Area\_fieldname]

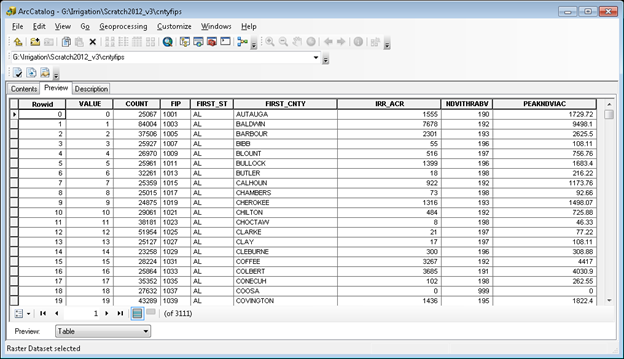
Example of the python process run for the 2012 data using Python 32-bit version from ArcGIS 10.2



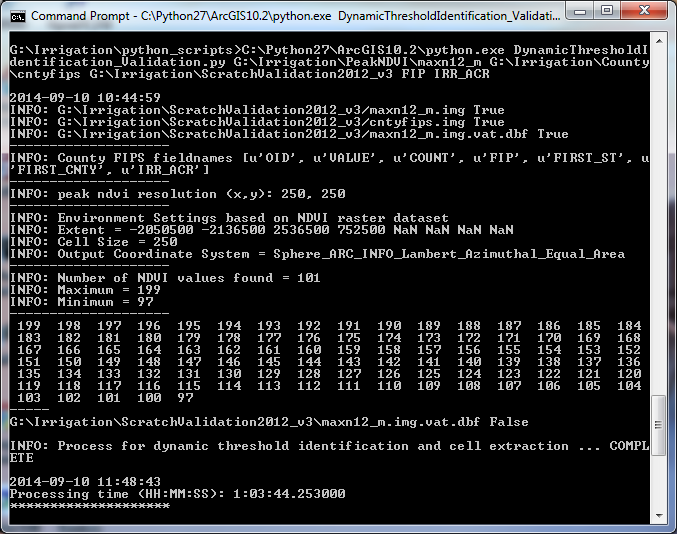
Example of the python process run for the 2012 data using Python 32-bit version from ArcGIS 10.2



### Output for 2012 python process



Example of the python process run for the 2012 data using Python 32-bit version from ArcGIS 10.2



### Output for 2012 python process for validation purposes

