

GEOG 432/832: Programming, Scripting, and Automation for GIS

Unit 08.01: Rasters

Dr. Bitterman

Today's schedule

- Open discussion
- Slides, discussion and exercises
 - Today: reading raster data sets and interrogating the data structure
 - Wednesday: raster algebra and more
- For next class

Open discussion

How's lab 3 going?

Projects?

Raster data structure review

1. What is a raster dataset?
2. How are they used?
3. What are the key properties of a raster?
4. Any other concerns (e.g., topology) that you're aware of?

Raster processing in ArcPy

- Requires the Spatial Analyst module (`arcpy.sa.*`)
- Which requires the proper license (check your textbook for how to do it programmatically)
- Your book likes to do the following:

```
import arcpy
from arcpy.sa import *

#then you can call functions directly, for example
outraster = Slope(elev)
```

I am not a fan, I prefer to be more explicit (but you do you)

```
#my way
outraster = arcpy.sa.Slope(elev)
```

Download the *week08inclass.zip* file from Canvas

- Start a new project
- Extract the data
- Create a new arcpy notebook

Let's start with listing our data

- How have we done this in the past?
- Listing rasters is pretty similar

```
wkspc_string = "C:\\Users\\pjbitterman\\Dropbox\\GE0G432\\week8_rasters\\unit08_data"
arcpy.env.workspace = wkspc_string

arcpy.ListRasters()
```

Give it a shot - what were the results?

Let's read one

How is reading rasters different than reading feature classes?

```
# read
myraster = arcpy.Raster("nlcd_lc_14n")
# display
myraster
```

What happened?

How do we learn something about our datasets?

We can "describe" our rasters, just like feature classes

For example...

before you try it, what's the output?

```
mydesc = arcpy.da.Describe(myraster)  
mydesc
```

Checking raster properties

```
print(mydesc["dataType"])
print(mydesc["bandCount"])
print(mydesc["compressionType"])
print(mydesc["height"])
print(mydesc["width"])
print(mydesc["meanCellHeight"])
print(mydesc["meanCellWidth"])
print(mydesc["pixelType"])
print(mydesc["isInteger"])
print(mydesc["noDataValue"])
```

Try a few

We can also access cells directly to determine their values:

Let's break it down: what does the code do?

```
for x, y in myraster:  
    if(x < 10 and y < 10):  
        print(x, y, myraster[x, y])
```

What happened?

Modifying values directly

As always, let's break it down

```
myraster.save("nlcd_test")
testraster = arcpy.Raster("nlcd_test")
testraster.readOnly = False

for x, y in testraster:
    if testraster[x,y] == 41 or testraster[x,y] == 81:
        testraster[x,y] = 71

testraster.save()
```

What happened? How can you check it worked?

Using "canned" geoprocessing functions

- Let's use `ned30lc.tif` to develop a raster that describes the slope of the landscape
- What is `ned30lc.tif` ? How would you know?

Calculating slope from a DEM

```
in_dem = arcpy.Raster("ned30lc.tif")  
  
outraster = arcpy.sa.Slope(in_dem)  
outraster.save("myslope.tif")
```

What happened?

For next class

- Read Chapter 10 for this week