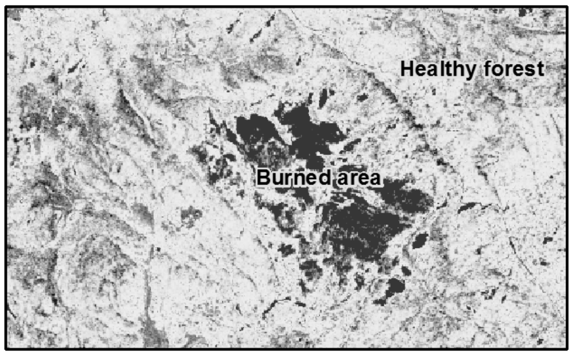
GIS Programming: Assignment 5
Vegetation Recovery after Wildfire

Due October 31 (30 points)

Scenario: You have been contracted by the U.S. Forest Service to model vegetation recovery after the Big Elk wildfire, which burned 4,413 acres between Lyons and Estes Park in July 2002. Your task is to analyze the relationship between recovery and terrain, specifically slope and aspect. As an independent contractor with a small budget you do not have access to an ArcGIS license, so you will have to conduct the analysis with open source tools.



Data: Extract the *lab5.zip* file from Canvas into your working directory. The data folder contains the following: a subfolder named *L5_big_elk*, which contains bands 3 and 4 (the red and near infrared bands, respectively) for a series of Landsat 5 Thematic Mapper (TM) images from 2002 to 2011 (all images were captured between August and October); a raster delineating the fire perimeter and an area of healthy forest outside of the fire perimeter (values of 2 indicate healthy forest, values of 1 indicate burned pixels, and values of 255 indicate a transition zone between the two that will be excluded from the analysis); and a DEM.

Goal: You will use the open source module Rasterio for input/output (I/O) to work with spatial data in Python. Building on your experience of Numpy and Scipy, you will use these tools for spatiotemporal raster analysis. You will work with some helper functions that I have prepared for you and build on these tools to develop your own open source solutions. You will write your own zonal statistics function in this lab and learn to use the Pandas module.

Overview: The Normalized Difference Vegetation Index (NDVI) can be used to measure relative abundance of green vegetation and biomass production. NDVI is calculated from the red and near infrared bands of a remotely sensed image. For Landsat 5, use the following:

$$NDVI = \frac{Band\ 4 - Band3}{Band4 + Band3}$$

Environmental conditions change from year to year and season to season. So, comparing biomass production and abundance between two points in time with satellite imagery comes with a significant amount of uncertainty. Variation in atmospheric conditions, scene illumination and sensor viewing geometry further complicate temporal comparison. In light of this, you will calculate a Recovery Ratio (RR) between the NDVI of each burned pixel inside the fire perimeter and the mean NDVI of healthy forest pixels outside of the burned area. This allows you to estimate vegetation recovery after the fire while minimizing the aforementioned issue of simply comparing NDVI across the years of analysis. Once you have calculated the RR for each pixel inside the fire perimeter, you will calculate the coefficient of recovery of the RR for each pixel across the years of analysis by fitting a first-degree polynomial. You will then summarize the vegetation recovery by terrain slope and aspect using a Zonal Statistics operation.

Part I:

1. Read the DEM in as a numpy array and, using the functions provided, calculate the slope and aspect. Reclassify the aspect grid to **8** cardinal directions (N, NE, E, SE, etc.). Reclassify the slope grid into **10** classes using the function provided.
2. Calculate the Normalized Difference Vegetation Index (NDVI) for all years of analysis from the Landsat images.
3. Calculate the Recovery Ratio for each pixel for each year:

$$RR = \frac{NDVI_{ij}}{\text{mean(NDVI of healthy forest)}}$$

   Remember that the denominator is the mean NDVI of all healthy forest pixels for a given year (i.e. for each pixel_{ij} in a given year, you will use the same number in the denominator). HINT: you can get the denominator using Boolean indexing to subset the values you need.
4. Calculate the trend of the RR for each pixel across the years of analysis. HINT: use polyfit from Scipy to fit a first-degree polynomial. This fits a least squares trend line. Imagine the 10 RR values for each pixel on the *y* axis and 10 sequenced integers (2002-2011) on the *x* axis. Fit a line through the points. The polyfit() function returns two values, the first is the slope of the line (this is the **coefficient of recovery**), and the second is the intercept (you don't need this).
5. Print the mean RR for each year and also print the mean **coefficient of recovery** across all years for the burned area (remember, you need to exclude the values outside of the burned area in these calculations).

Part II:

1. Write a generic function that calculates "Zonal Statistics as Table" using two numpy arrays (the zone raster and the value raster). Your function should calculate: mean, standard deviation, min max and count. Use Pandas to create a *dataframe* with the zonal output and write the results to a csv file. The zonal output csv file should have the zone field (slope or aspect classes) and the five statistics (min, max, mean, stddev, and count). Your output csv table should have as many rows as there are unique zone values.
2. Calculate zonal stats of the **coefficient of recovery** for each **terrain slope** class **<u>and</u>** then for each **cardinal direction of aspect**, which you reclassified in Part I. You should produce two output csv files, one for **slope** and one for **aspect**. Since you are only interested in the burned pixels, your zonal stats function must somehow exclude all pixels outside of the fire perimeter. Hint: numpy.nan.
3. Export the final **coefficient of recovery** np array for the burned area as a GeoTiff (the extent should match that of the inputs). Non-burned pixels should have a NoData value (e.g. -99).
4. What are your conclusions regarding vegetation recovery and terrain? Provide a print statement explaining the relationship you found between slope and aspect of the terrain and vegetation recovery after wildfire.

I have provided you with a number of functions to get started; build upon this suite of open source GIS tools. As always, you should develop a solution that is logical, efficient, and scripted in a clear, readable manner.

Make the program as sophisticated as you can. This is our last lab and you have learned a great deal about Python, Numpy, Scipy and now Pandas. Develop generic solutions using advanced logic and techniques. Create functions to improve the flow of your program.

By the start of next class in two weeks upload your script (*lastname_lab5.py*) to Canvas.

Grading: You will be evaluated on the following: your script runs without errors (3), Part I: calculates the RR (8), calculates the coefficient of recovery (6), and prints the correct values (2); Part II: Zonal Stats function (4), final grid, zonal csv files, and conclusions are correct (2); the clarity, logic and efficiency of your script (5).

Challenge: Improve your Zonal Statistics function so that it accommodates a shapefile or raster as the zone data layer. Plot the mean RR across the years of analysis using the matplotlib library.