GEOG 577 Lab 2

**Due Feb 7**



**Search** for the NEON Bidirectional Surface Reflectance Dataset.

**Read** about the image collection and look through the attributes of the data (e.g. bands, properties, etc).

**Import** the image collection.

**Add** the image collection to the map layer. Also **print** it to the console.

1. **Write** a paragraph about the general characteristics of this dataset, such as spatial, spectral, radiometric resolutions, etc. Include some mention of the program under which the data has been collected and archived, as well as the geographic limits on the availability of the data.

**Select** out the “Northern Great Plains Research Laboratory” and save that as its own image. A couple of **hints** here: **a)** Image collections have ‘properties’ that can be used to filter the data using .filterMetadata. In this case the NEON image data has a property called 'NEON\_SITE\_NAME' (among many other properties) and you want to retrieve only the data for which that property equals ‘Northern Great Plains Research Laboratory’ . I hope that makes sense. Images inherit the properties from their parent image collections. In case you were wondering. **b)** One way to render an “image collection” to an “image” is to use the .mosaic() method, as we’ve seen in a prior lab. Another way is to use the .first() or .last() methods. Either one will work here.There are also other ways . . .

**Use** Map.centerObject to set your map display to the location of the image and zoom in to about 12.

**Display** the data and render it using a B364, B091, B008 composite and apply a 98% or 100% stretch so you can see what this data looks like.

1. **Where** is this?!
2. **What** wavelength regions to those three bands correspond to?
3. **Zoom** to the base resolution of the data and explore different regions of the imagery. What ‘Zoom Level’ corresponds to the base resolution of the data?
4. **Make** a casual list of the different cover types, land uses and surface types that you observe in this image.

As an aside let’s incorporate some code that will generate spectral signatures for any image pixel that we click on. It’s a bit cumbersome and opaque, and the methods are somewhat specific to this dataset, so below in blue is a bunch of code that will help. It will need some tinkering to make it work for you. At the very least you’ll need to adjust for filenames and any characters that don’t translate if you copy and paste. Fair warning. Here goes:

// Pull out only the data bands (these all start with B, eg. B001)

var NGPData = NGP.select('B.\*')

print('NGP brdf data', NGPData) //This is a good step so that you can look in //the console and see what you got.

// Read in the properties as a dictionary. Just trust me.

var properties = NGP.toDictionary()

// Select the WL\_FWHM\_B\*\*\* band properties (using regex). Trust me.

var wl\_fwhm\_dict = properties.select(['WL\_FWHM\_B+\\d{3}']);

// Pull out the wavelength, fwhm values to a list. jtm.

var wl\_fwhm\_list = wl\_fwhm\_dict.values()

print('Wavelength FWHM list:',wl\_fwhm\_list)

// Function to pull out the wavelength values only and convert the string to //float. ™, really.

var get\_wavelengths = function(x) {

var str\_split = ee.String(x).split(',')

var first\_elem = ee.Number.parse((str\_split.get(0)))

return first\_elem

}

// apply the function to the wavelength full-width-half-max list

var wavelengths = wl\_fwhm\_list.map(get\_wavelengths)

print('Wavelengths:',wavelengths)

print('Number of data bands:',wavelengths.length()) //worth taking a look at this.

// Create a panel to hold the spectral signature plot

var panel = ui.Panel();

panel.style().set({width: '600px',height: '300px',position: 'top-left'});

Map.add(panel);

Map.style().set('cursor', 'crosshair');

// Create a function to draw a chart when a user clicks on the map.

Map.onClick(function(coords) {

panel.clear();

var point = ee.Geometry.Point(coords.lon, coords.lat);

wavelengths.evaluate(function(wvlnghts) {

var chart = ui.Chart.image.regions({

image: NGPData,

regions: point,

scale: 1,

seriesProperty: 'λ (nm)',

xLabels: wavelengths.getInfo()

});

chart.setOptions({

title: 'Reflectance',

hAxis: {title: 'Wavelength (nm)',

vAxis: {title: 'Reflectance'},

gridlines: { count: 5 }}

});

// Create and update the location label

var location = 'Longitude: ' + coords.lon.toFixed(2) + ' ' +

'Latitude: ' + coords.lat.toFixed(2);

panel.widgets().set(1, ui.Label(location));

panel.add(chart);

})

});

1. **Explain** (briefly) how you solved any problems encountered in getting the above script to work.

Finally see if you can create a co-spectral plot. Make a plot of red (x-axis) on nir (y-axis) reflectance for this scene.

1. **Which** band numbers did you use (or intend to use)?
2. **How** did you figure out how to do this (in one or two sentences)?