Abib-Nicole-rgbNotebook

June 30, 2018

0.1 Pre-Institute Week 3: Working with RGB Imagery in Python

In [1]:
```
# Check which version of python is running
import sys
sys.version
```

Out[1]:
```
'3.5.5 |Anaconda custom (64-bit)| (default, Apr 26 2018, 08:11:22) \n[GCC 4.2.1 Compat
```

In [2]:
```
# Import required packages
import gdal
```

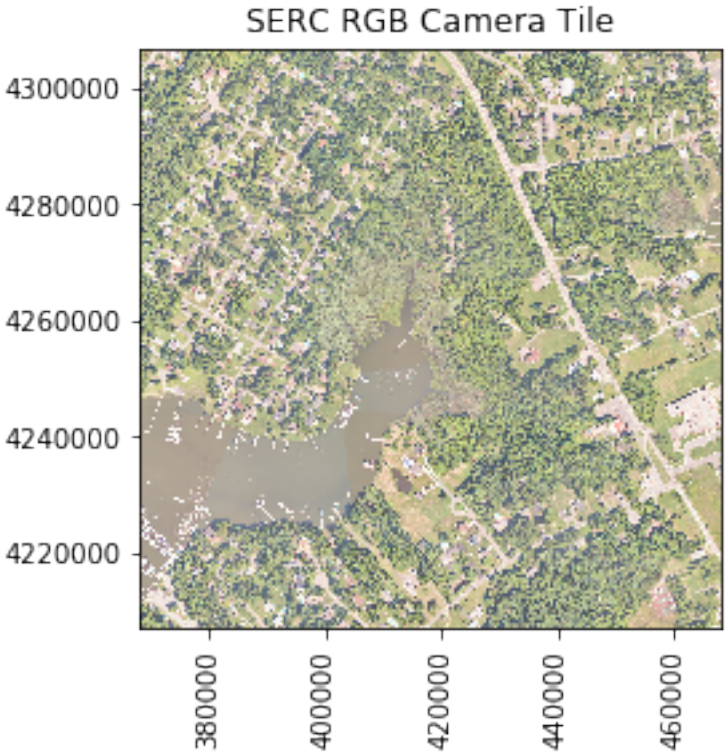
In [3]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```
# Convert RGB raster to a Numpy Array

def RGBraster2array(RGB_geotif):
    """RGBraster2array reads in a NEON AOP geotif file and returns
    a numpy array, and header containing associated metadata with spatial information.
    --------
    Parameters
        RGB_geotif -- full or relative path and name of reflectance hdf5 file
    --------
    Returns
    --------
    array:
        numpy array of geotif values
    metadata:
        dictionary containing the following metadata (all strings):
            array_rows
            array_cols
            bands
            driver
            projection
            geotransform
```

```
        pixelWidth
        pixelHeight
        extent
        noDataValue
        scaleFactor
    --------
    Example Execution:
    --------
    RGB_geotif = '2017_SERC_2_368000_4306000_image.tif'
    RGBcam_array, RGBcam_metadata = RGBraster2array(RGB_geotif) """

    metadata = {}
    dataset = gdal.Open(RGB_geotif)
    metadata['array_rows'] = dataset.RasterYSize
    metadata['array_cols'] = dataset.RasterXSize
    metadata['bands'] = dataset.RasterCount
    metadata['driver'] = dataset.GetDriver().LongName
    metadata['projection'] = dataset.GetProjection()
    metadata['geotransform'] = dataset.GetGeoTransform()

    mapinfo = dataset.GetGeoTransform()
    metadata['pixelWidth'] = mapinfo[1]
    metadata['pixelHeight'] = mapinfo[5]

    metadata['ext_dict'] = {}
    metadata['ext_dict']['xMin'] = mapinfo[0]
    metadata['ext_dict']['xMax'] = mapinfo[0] + dataset.RasterXSize/mapinfo[1]
    metadata['ext_dict']['yMin'] = mapinfo[3] + dataset.RasterYSize/mapinfo[5]
    metadata['ext_dict']['yMax'] = mapinfo[3]

    metadata['extent'] = (metadata['ext_dict']['xMin'],metadata['ext_dict']['xMax'],
                          metadata['ext_dict']['yMin'],metadata['ext_dict']['yMax'])

    raster = dataset.GetRasterBand(1)
    array_shape = raster.ReadAsArray(0,0,metadata['array_cols'],metadata['array_rows'])
    metadata['noDataValue'] = raster.GetNoDataValue()
    metadata['scaleFactor'] = raster.GetScale()

    array = np.zeros((array_shape[0],array_shape[1],dataset.RasterCount),'uint8') #pre
    for i in range(1, dataset.RasterCount+1):
        band = dataset.GetRasterBand(i).ReadAsArray(0,0,metadata['array_cols'],metadata
        band[band==metadata['noDataValue']]=np.nan
        band = band/metadata['scaleFactor']
        array[...,i-1] = band

    return array, metadata

In [5]: # Load geotif
```

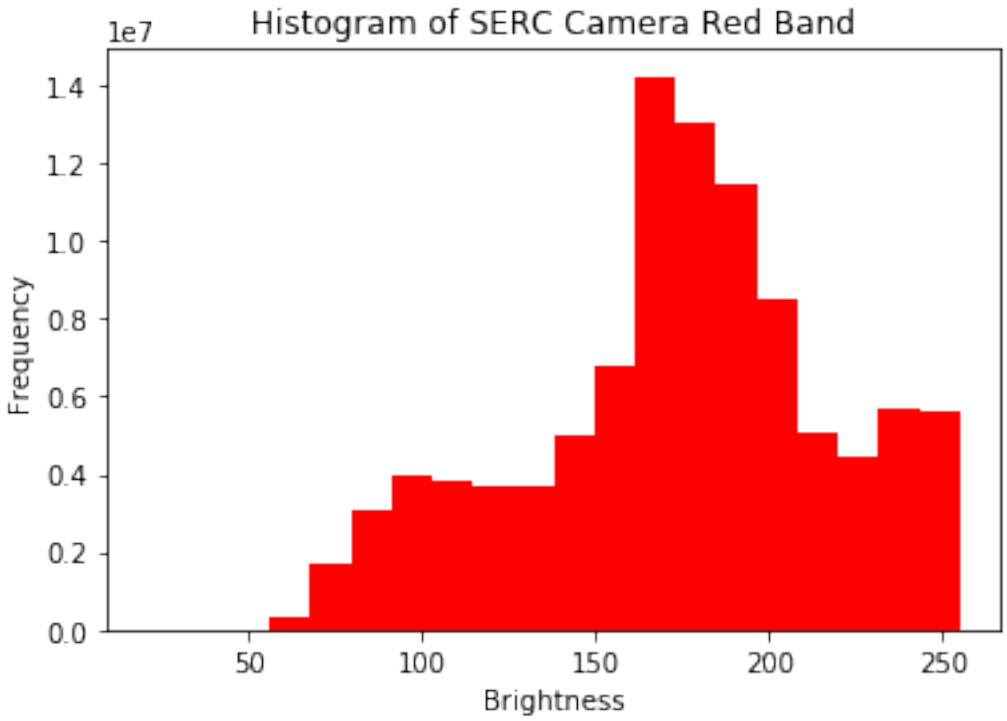
```
        RGB_geotif = '//Users/nabib/data/neon_aop_rgb/2017_SERC_2_368000_4306000_image.tif'
        SERC_RGBcam_array, SERC_RGBcam_metadata = RGBraster2array(RGB_geotif)

In [6]: # Check dimensions of array
        SERC_RGBcam_array.shape

Out[6]: (10000, 10000, 3)

In [7]: #Display information stored in header
        for key in sorted(SERC_RGBcam_metadata.keys()):
          print(key)

array_cols
array_rows
bands
driver
ext_dict
extent
geotransform
noDataValue
pixelHeight
pixelWidth
projection
scaleFactor


In [8]: # Define function to plot the array
        def plot_band_array(band_array,
                            refl_extent,
                            colorlimit,
                            ax=plt.gca(),
                            title='',
                            cbar ='on',
                            cmap_title='',
                            colormap='spectral'):

            '''plot_band_array reads in and plots a single band or an rgb band combination of
            --------
            Parameters
            --------
                band_array: flightline array of reflectance values, created from h5refl2array
                refl_extent: extent of reflectance data to be plotted (xMin, xMax, yMin, yMax)
                colorlimit: range of values to plot (min,max). Best to look at the histogram o
                ax: optional, default = current axis
                title: string, optional; plot title
                cmap_title: string, optional; colorbar title
                colormap: string, optional; see https://matplotlib.org/examples/color/colormap
            --------
            Returns
```

```
        plots array of single band or RGB if given a 3-band
    --------
    Example:
    --------
    plot_band_array(SERC_RGBcam_array,
                    SERC_RGBcam_metadata['extent'],
                    (1,255),
                    title='SERC RGB Camera Tile',
                    cbar='off')'''

    plot = plt.imshow(band_array,extent=refl_extent,clim=colorlimit);
    if cbar == 'on':
        cbar = plt.colorbar(plot,aspect=40); plt.set_cmap(colormap);
        cbar.set_label(cmap_title,rotation=90,labelpad=20)
    plt.title(title); ax = plt.gca();
    ax.ticklabel_format(useOffset=False, style='plain'); #do not use scientific notati
    rotatexlabels = plt.setp(ax.get_xticklabels(),rotation=90); #rotate x tick labels

# Plot our array
plot_band_array(SERC_RGBcam_array,
            SERC_RGBcam_metadata['extent'],
            (1,255),
            title='SERC RGB Camera Tile',
            cbar='off')
```



```
In [9]: # Plot histogram of red band
        plt.hist(np.ravel(SERC_RGBcam_array[:,:,0]),20,facecolor='red');
        plt.title('Histogram of SERC Camera Red Band')
        plt.xlabel('Brightness'); plt.ylabel('Frequency')
```

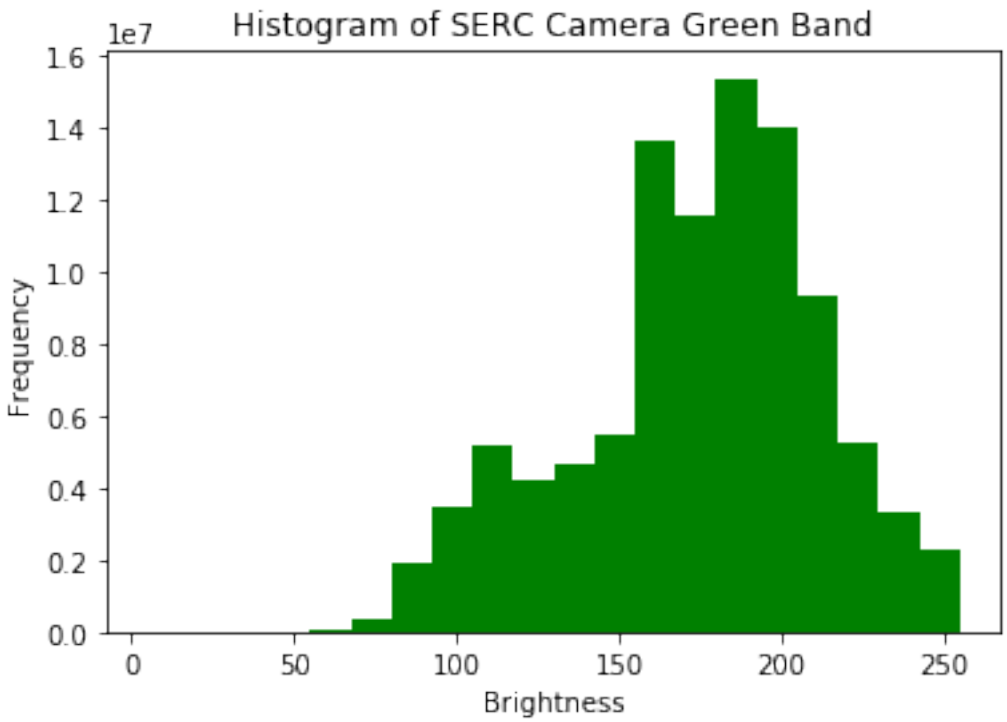
```
Out[9]: Text(0,0.5,'Frequency')
```



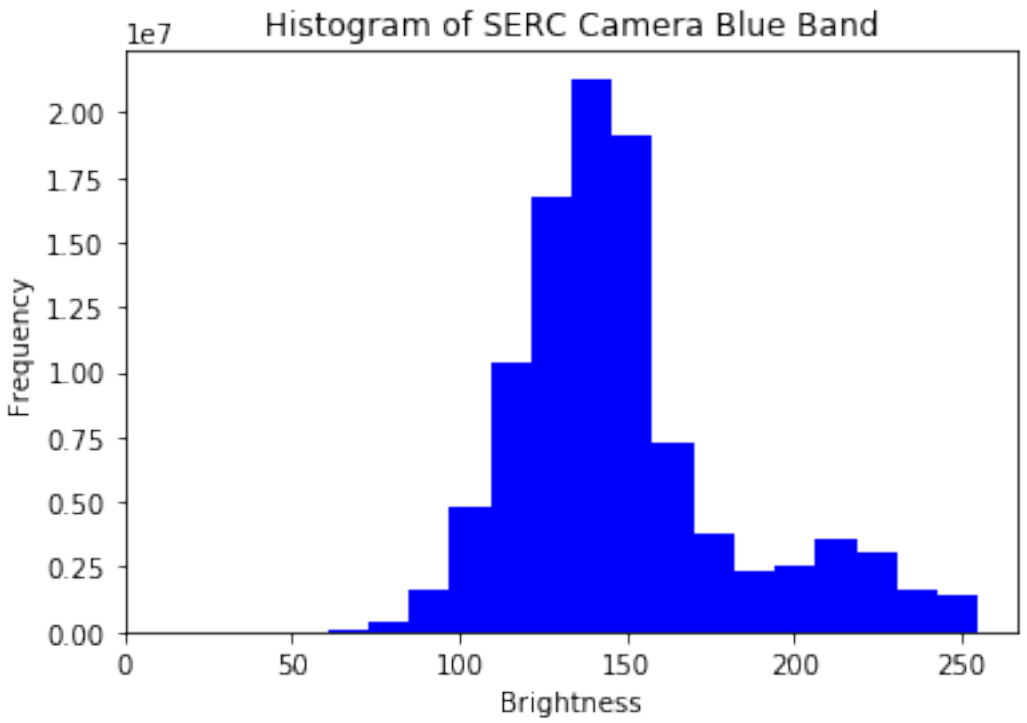
0.2 Challenge Exercises

```
In [10]: # Plot histogram of green band
         plt.hist(np.ravel(SERC_RGBcam_array[:,:,1]),20,facecolor='green');
         plt.title('Histogram of SERC Camera Green Band')
         plt.xlabel('Brightness'); plt.ylabel('Frequency')
```

```
Out[10]: Text(0,0.5,'Frequency')
```




```
In [11]: # Plot histogram of blue band
         plt.hist(np.ravel(SERC_RGBcam_array[:,:,2]),20,facecolor='blue');
         plt.title('Histogram of SERC Camera Blue Band')
         plt.xlabel('Brightness'); plt.ylabel('Frequency')

Out[11]: Text(0,0.5,'Frequency')
```

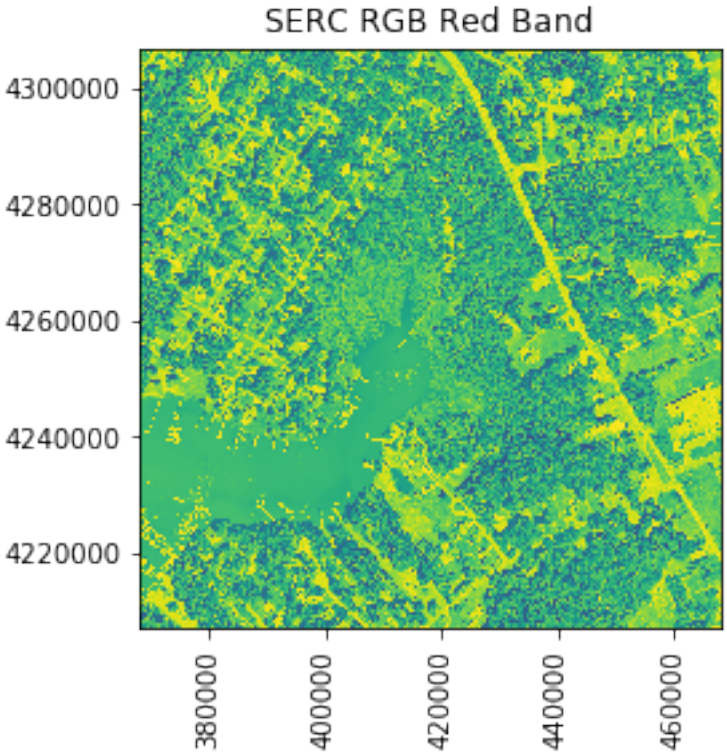



```
In [13]: # Extract data from the array
         red_min = np.amin(SERC_RGBcam_array[:,:,0]);
         red_max = np.amax(SERC_RGBcam_array[:,:,0]);
         print ('The minimum reflectance for Band 1 is:',red_min,'\nThe maximum reflectance fo

         green_min = np.amin(SERC_RGBcam_array[:,:,1]);
         green_max = np.amax(SERC_RGBcam_array[:,:,1]);
         print ('\nThe minimum reflectance for Band 2 is:',green_min,'\nThe maximum reflectanc

         blue_min = np.amin(SERC_RGBcam_array[:,:,2]);
         blue_max = np.amax(SERC_RGBcam_array[:,:,2]);
         print ('\nThe minimum reflectance for Band 3 is:',blue_min,'\nThe maximum reflectance

         print('\nThe projection is:',SERC_RGBcam_metadata['projection'])
```

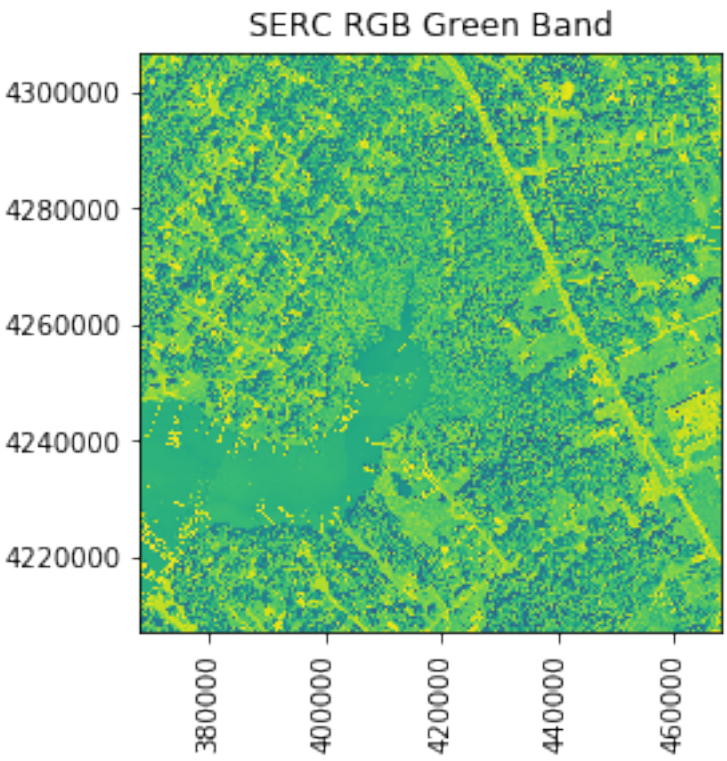
```
The minimum reflectance for Band 1 is: 21
The maximum reflectance for Band 1 is: 255

The minimum reflectance for Band 2 is: 5
The maximum reflectance for Band 2 is: 255

The minimum reflectance for Band 3 is: 12
The maximum reflectance for Band 3 is: 255
```

```
The projection is: PROJCS["WGS 84 / UTM zone 18N",GEOGCS["WGS 84",DATUM["WGS_1984",SPHEROID["W
```

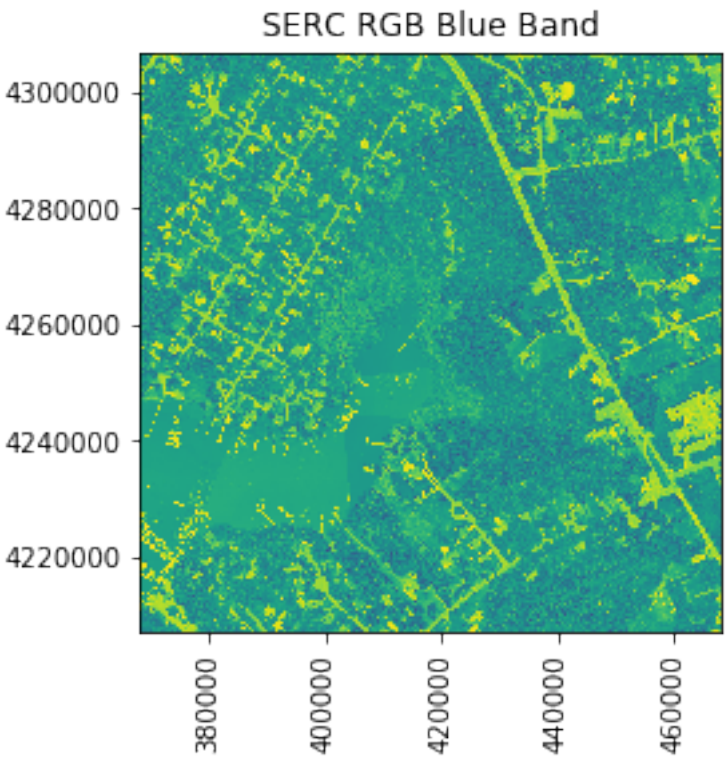
```
In [14]: # Plot the red band
         plot_band_array(SERC_RGBcam_array[:,:,0],
                         SERC_RGBcam_metadata['extent'],
                         (1,255),
                         title='SERC RGB Red Band',
                         cbar='off')
```




```
In [15]: # Plot the green band
         plot_band_array(SERC_RGBcam_array[:,:,1],
                         SERC_RGBcam_metadata['extent'],
                         (1,255),
                         title='SERC RGB Green Band',
                         cbar='off')
```



In [16]:
```
# Plot the blue band
plot_band_array(SERC_RGBcam_array[:,:,2],
                SERC_RGBcam_metadata['extent'],
                (1,255),
                title='SERC RGB Blue Band',
                cbar='off')
```


SERC RGB Blue Band
4300000
4280000
4260000
4240000
4220000
380000
400000
420000
440000
460000