

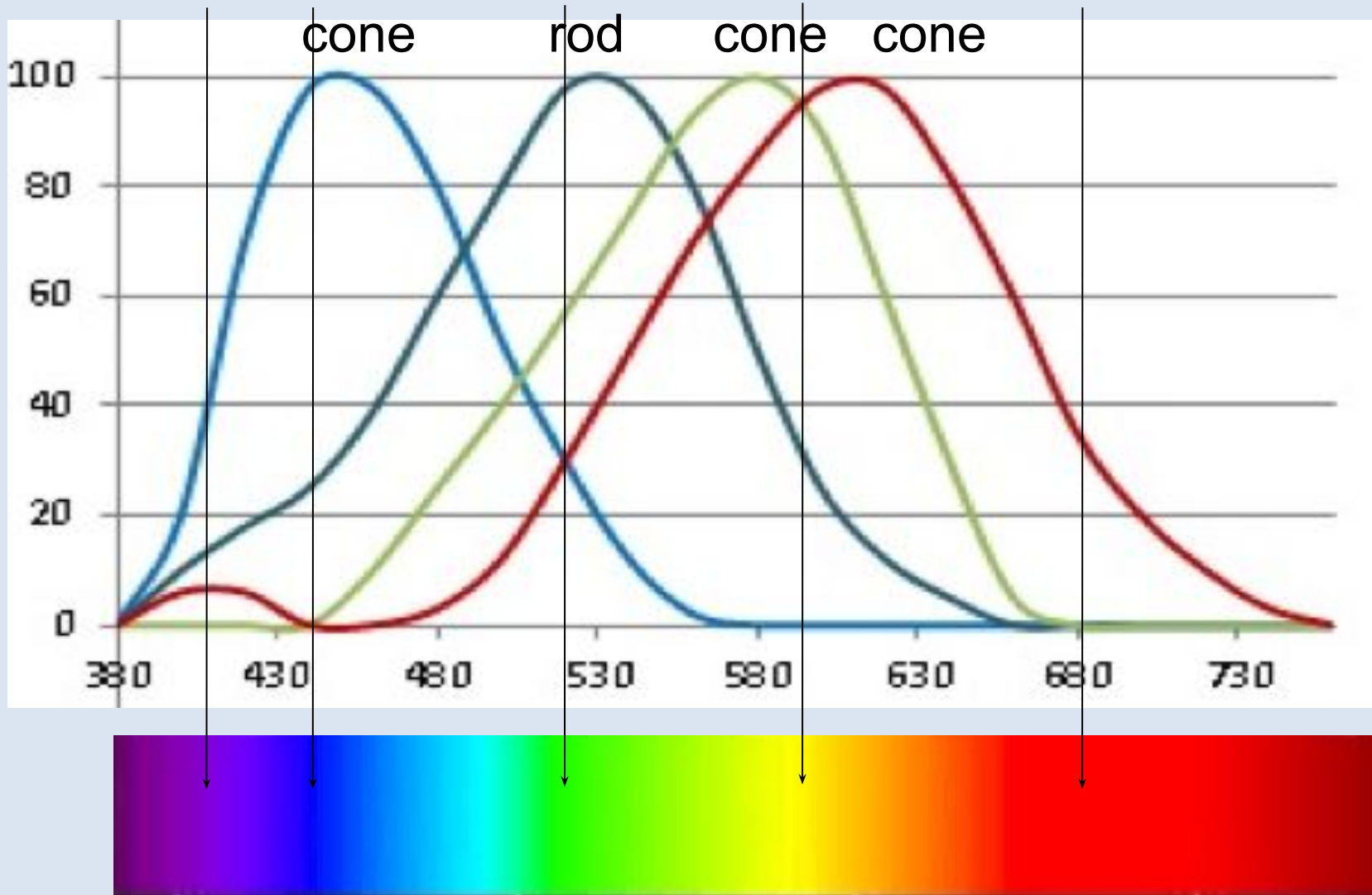
Satellite Imagery

Getting Started with Python

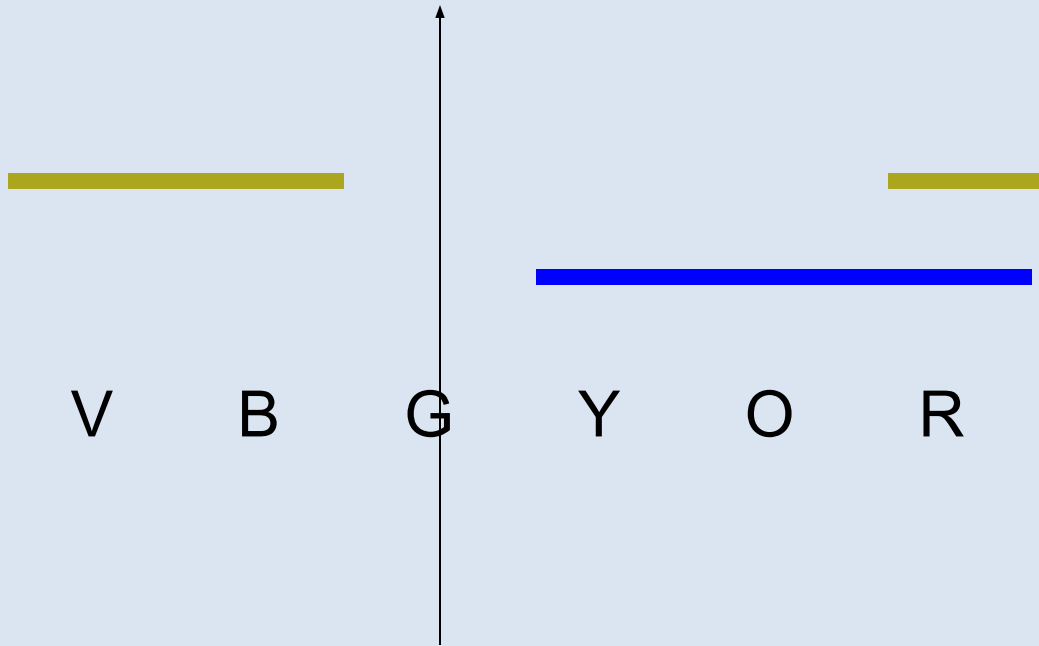
Outline

- Creating RGB and grey scale images
- Data Formats : text, binary, self-descriptive
- Making RGB images from Landsat data
- Vegetation Indices and maps

Rods, Cones, (and purple)

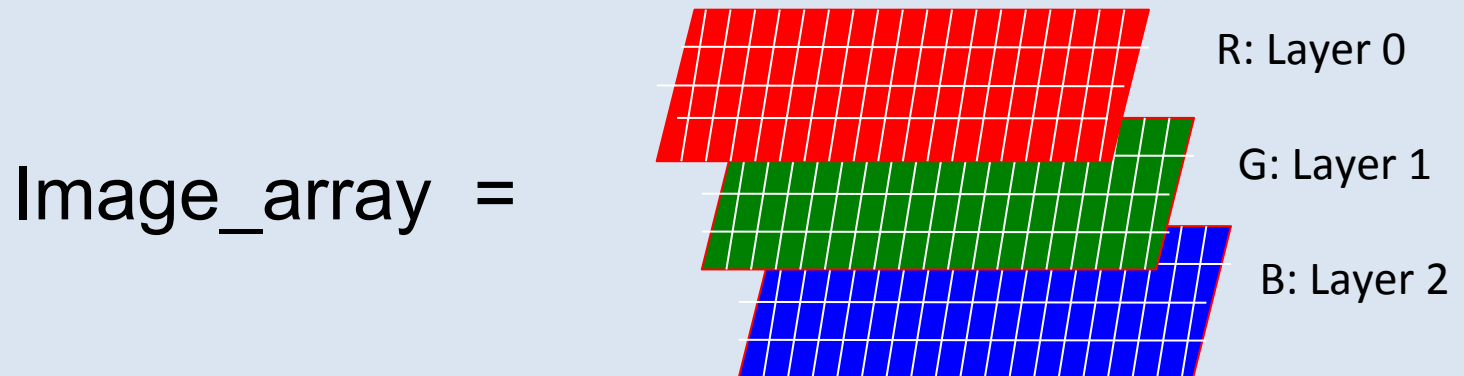


Subtractive Colors (paint)



Making Color Images in Python

`Image_array = (rows, cols, 3 layers)`



The values in each layer must range from 0 to 1, or they will be “railed off”. You may need to adjust your data accordingly.

Color Programming Exercises

```
>>> import numpy as np # well duh >>> plt.on()
```

Make a zero-filled array called RGB with 300 rows, 300 columns, 3 layers.

```
>>> RGB = np.zeros([300,300,3],dtype=float)
```

Display this image: what color should it be?

```
>>> plt.imshow(RGB) >>> plt.show()
```

Turn the image Red.

```
>>> RGB[:, :, 0] = 1 >>> plt.imshow(RGB)
```

Turn the image White.

```
>>> RGB[:, :, :] = 1 >>> plt.imshow(RGB)
```

Turn the image Grey.

```
>>> RGB[:, :, :] = 0.5 >>> plt.imshow(RGB)
```

Color Exercises Continued

Turn the center third of the image Green.

```
>>> RGB[ 100:200 , 100:200 , 1 ] = 1           >>> plt.imshow(RGB)
```

Turn the center third of the image Yellow.

```
>>> RGB[ 100:200 , 100:200 , 0 ] = 1           >>> plt.imshow(RGB)
```

Make the upper left corner Blue and lower right corner Red.

```
>>> RGB[ 0:100 , 0:100 , 2 ] = 1  
>>> RGB[200:300,200:300,0 ] = 1  
>>> plt.imshow(RGB)  
>>> plt.savefig( 'myfigure.png' ) # saves figure
```

Basic Data Formats

Text - always ASCII, arranged in rows, columns, headers.

NEVER word processing format or excel. '.csv' is text.

PRO – easy for humans to read and understand.

CON – slow for computer to read, uses more storage than any other format. Easy to corrupt during viewing.

USE – small files. More convenient for humans.

Binary - single vector with each field of a known size.

Will need to rearrange into a grid after reading.

PRO – easy and fast for computer to read. Very compact.

CON – humans can only read through writing an interpreter.

Requires extra information into order to read.

USE – large files. Replaced by self-describing files for satellite.

Self Descriptive Formats

The data is encoded as binary for speed and compactness, but has descriptors embedded in the file so that special “readers” (packages of computer code) can pull out the “metadata” to format and display the data.

geoTIFF - a TIFF image with extra geographical and data set descriptors embedded. No reader needed to see image.

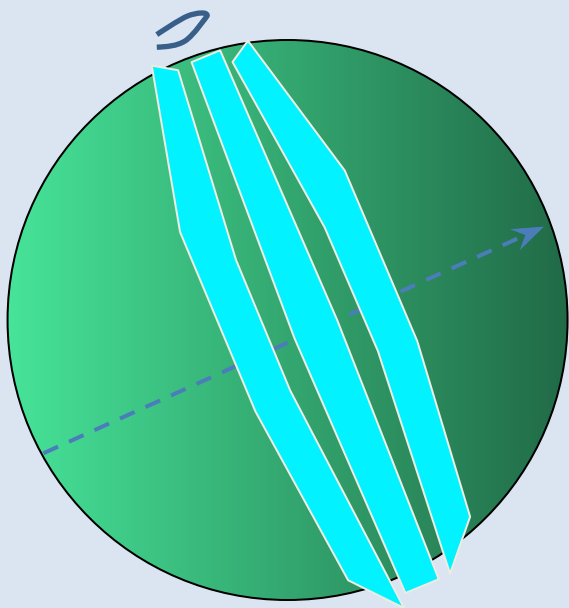
USE – primarily Landsat and other USGS products.

netCDF - “network Common Data Format” created at the request of NASA (which later switched mainly to HDF)

USE – the majority of NOAA satellite products and some by NASA, which is now providing multiple formats

HDF - “Hierarchical Data Format” more flexible but more complex than netCDF.

USE – mainly by NASA-EOS, but the new HDF5 format incorporates netCDF4, and is becoming more standard.



The Landsat Satellite

Due to the narrow swath of 185 km, the revisit period is 16 days, though overlaps may happen every week. Overpass time ~10 am. Data begins in 1978.

Landsat 8 Operational Land Imager (OLI) and Thermal Infrared Sensor (TIRS) images consist of nine spectral bands with a spatial resolution of 30 meters for Bands 1 to 7 and 9. New band 1 (ultra-blue) is useful for coastal and aerosol studies. New band 9 is useful for cirrus cloud detection. The resolution for Band 8 (panchromatic) is 15 meters. Thermal bands 10 and 11 are useful in providing more accurate surface temperatures and are collected at 100 meters. Approximate scene size is 170 km north-south by 183 km east-west (106 mi by 114 mi).

Landsat 8 Operational Land Imager (OLI) and Thermal Infrared Sensor (TIRS) Launched February 11, 2013	Bands	Wavelength (micrometers)	Resolution (meters)
	Band 1 - Coastal aerosol	0.43 - 0.45	30
	Band 2 - Blue	0.45 - 0.51	30
	Band 3 - Green	0.53 - 0.59	30
	Band 4 - Red	0.64 - 0.67	30
	Band 5 - Near Infrared (NIR)	0.85 - 0.88	30
	Band 6 - SWIR 1	1.57 - 1.65	30
	Band 7 - SWIR 2	2.11 - 2.29	30
	Band 8 - Panchromatic	0.50 - 0.68	15
	Band 9 - Cirrus	1.36 - 1.38	30
	Band 10 - Thermal Infrared (TIRS) 1	10.60 - 11.19	100 * (30)
Band 11 - Thermal Infrared (TIRS) 2	11.50 - 12.51	100 * (30)	

* TIRS bands are acquired at 100 meter resolution, but are resampled to 30 meter in delivered data product.

We will assume Atmospheric Correction is applied



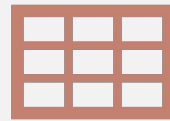
If you want to look at anything other than the atmosphere, you must subtract out the effect of the atmosphere. We must account for extinction and 'path radiance' (light scattered into the beam).

Making an RGB image with Landsat

Start with
landsat_main.py
And
landsat_module.py



Read the red, green,
blue, NIR data



Put into a 3-layer
RGB array



Display the image

Is Greenness the best Vegetation Indicator?



- Is anything else green?
- Is there some characteristic more sensitive to vegetation than green?

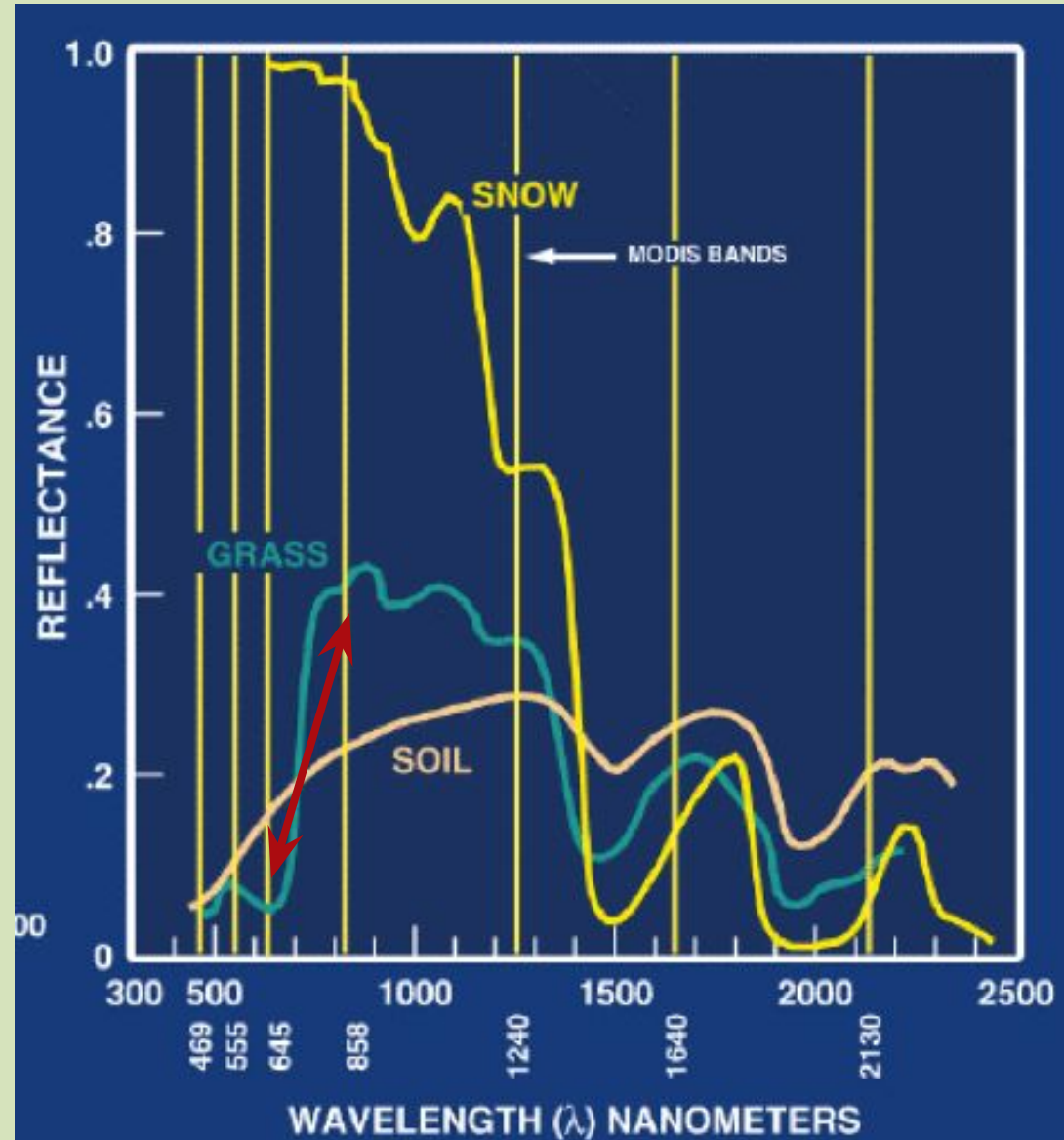
Normalized Difference Vegetation Index

High absorption in the red with high reflectance in the near infrared (NIR) is characteristic of chlorophyll.

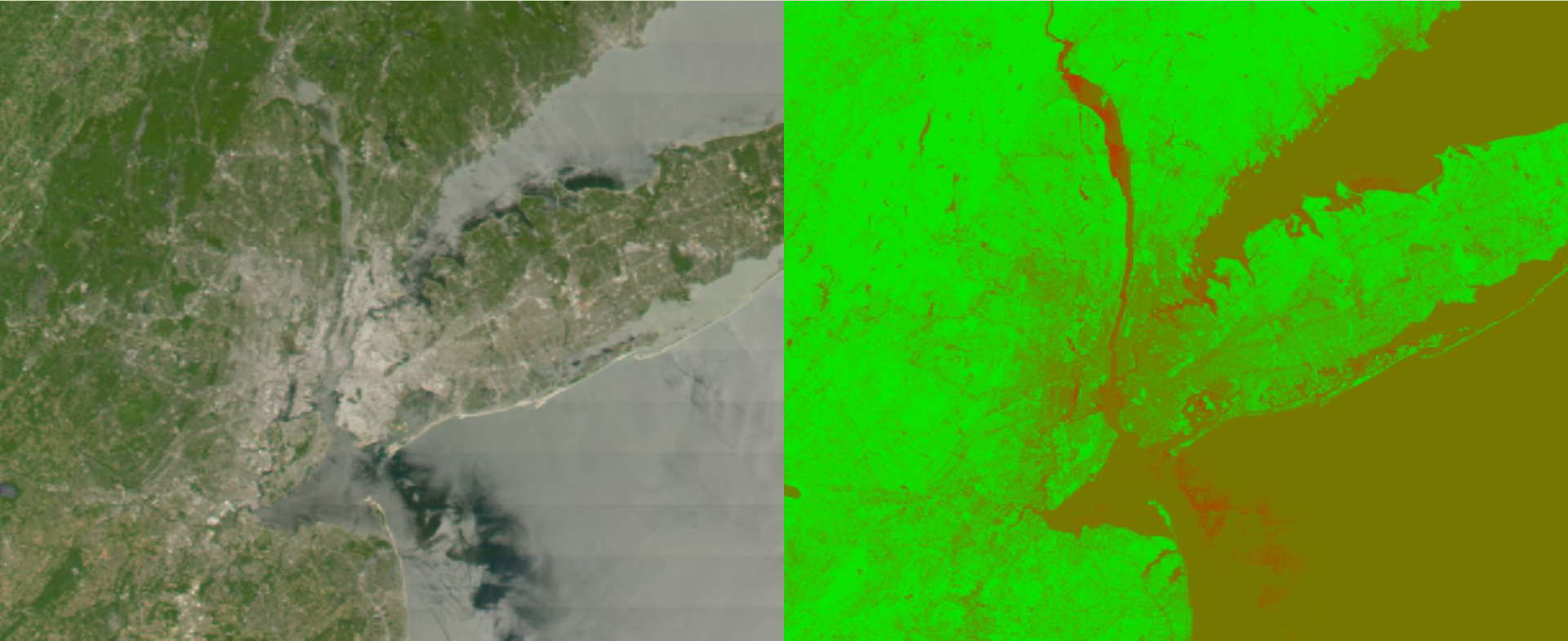
$$NDVI = \frac{NIR - RED}{NIR + RED}$$

$$-1 < NDVI < 1$$

High NDVI indicates vegetation density



Vegetation by Satellite



Visible RGB

500 m resolution

NDVI: red is negative

250 m resolution

Making a Landsat image of NDVI

- Calculate NDVI from NIR and Red (that we already have)
- Put NDVI into the green layer of an RGB array of the same horizontal dimensions
- Display the image

Exercise 1

Write a function that will display a map of NDVI in shades of green. It must have comments that explain the input and output as well as any internal calculations or variables. It must work when tested.

INPUT VARIABLES: Red reflectance, Near Infrared reflectance. Both in array format.

OUTPUT VARIABLES: NDVI in array format

OUTPUT PRODUCTS: map of NDVI, saved as .png file
~~map of RGB, saved as .png file~~