Vegetation indices in R

Learning Objectives

After completing this tutorial, you will be able to:

- Calculate NDVI and NBR in R
- Describe what a vegetation index is and how it is used with spectral remote sensing data.

What you need

You will need a computer with internet access to complete this lesson and the data for week 6 of the course.

[Download Week 6 Data (~500 MB)](

About vegetation indices

A vegetation index is a single value that quantifies vegetation health or structure. The math associated with calculating a vegetation index is derived from the physics of light reflection and absorption across bands. For instance, it is known that healthy vegetation reflects light strongly in the near infrared band and less strongly in the visible portion of the spectrum. Thus, if you create a ratio between light reflected in the near infrared and light reflected in the visible spectrum, it will represent areas that potentially have healthy vegetation.

Normalized difference vegetation index (NDVI)

The Normalized Difference Vegetation Index (NDVI) is a quantitative index of greenness ranging from 0-1 where 0 represents minimal or no greenness and 1 represents maximum greenness.

NDVI is often used for a quantitate proxy measure of vegetation health, cover and phenology (life cycle stage) over large areas.

NDVI is calculated from the visible and near-infrared light reflected by vegetation. Healthy vegetation (left) absorbs most of the visible light that hits it, and reflects a large portion of near-infrared light. Unhealthy or sparse vegetation (right) reflects more visible light and less near-infrared light. Source: NASA

- More on NDVI from NASA

Calculate NDVI

Sometimes we are able to download already calculated NDVI data products. In this case, we need to calculate NDVI ourselves using the reflectance data that we have.

```
# load spatial packages
library(raster)
library(rgdal)
library(rgeos)
library(RColorBrewer)
# turn off factors
options(stringsAsFactors = F)
```

```r
all_landsat_bands <- list.files("data/week6/Landsat/LC80340322016205-SC20170127160728/crop",
           pattern=glob2rx("*band*.tif$"),
           full.names = T) # use the dollar sign at the end to get all files that END WITH
all_landsat_bands

# stack the data
landsat_stack_csf <- stack(all_landsat_bands)
```

Calculate NDVI

The normalized difference vegetation index (NDVI) uses a ratio between near infrared and red light within the electromagnetic spectrum. To calculate NDVI we use the following formula where NIR is near infrared light and red represents red light. For our raster data, we will take the reflectance value in the red and near infrared bands to calculate the index. . `(NIR - Red) / (NIR + Red)`

```r
# calculate NDVI
landsat_ndvi <- (landsat_stack_csf[[5]] - landsat_stack_csf[[4]]) / (landsat_stack_csf[[5]] + landsat_s

plot(landsat_ndvi,
     main="Landsat derived NDVI\n 23 July 2016")
```

View distribution of NDVI values

```r
# view distribution of NDVI values
hist(landsat_ndvi,
  main="NDVI: Distribution of pixels\n Landsat 2016 Cold Springs fire site",
  col="springgreen")
```

Export raster

When you are done, you may want to export your rasters so you could use them in QGIS or ArcGIS or share them with your colleagues. To do this you use the writeRaster() function.

```r
# export raster
# NOTE: this won't work if you don't have an outputs directory in your week6 dir!
writeRaster(x = landsat_ndvi,
              filename="data/week6/outputs/landsat_ndvi.tif",
              format = "GTiff", # save as a tif
              datatype='INT2S', # save as a INTEGER rather than a float
              overwrite = T)  # OPTIONAL - be careful. this will OVERWRITE previous files.
```

Calculate Normalized Burn Ratio (NBR)

The Normalized burn ratio (NBR) highlights burned areas in large fire zones greater than 500 acres. The formula is similar to a normalized difference vegetation index (NDVI), except that it uses near-infrared (NIR) and shortwave-infrared (SWIR) wavelengths (Lopez, 1991; Key and Benson, 1995).

NBR = ((NIR - SWIR)/ (NIR + SWIR)) * 1000

The NBR was originally developed for use with Landsat TM and ETM+ bands 4 and 7, but it will work with any multispectral sensor (including Landsat 8) with a NIR band between 760 - 900 nm and a SWIR band between 2080 - 2350 nm.

Looking at the table below, what bands should we use for Landsat 8?

Landsat 8 Bands

Band	Wavelength range (nanometers)	Spatial Resolution (m)	Spectral Width (nm)
Band 1 - Coastal aerosol	430 - 450	30	2.0
Band 2 - Blue	450 - 510	30	6.0
Band 3 - Green	530 - 590	30	6.0
Band 4 - Red	640 - 670	30	0.03
Band 5 - Near Infrared (NIR)	850 - 880	30	3.0
Band 6 - SWIR 1	1570 - 1650	30	8.0
Band 7 - SWIR 2	2110 - 2290	30	18
Band 8 - Panchromatic	500 - 680	15	18
Band 9 - Cirrus	1360 - 1380	30	2.0

When you have calculated NBR - classify the output raster using the `classify()` function and the classes below.

SEVERITY LEVEL
Enhanced Regrowth
Unburned
Low Severity
Moderate Severity
High Severity

NOTE: your min an max values for NBR may be slightly different from the table shown above! If you have a smaller min value (< -700) then adjust your first class to that smallest number. If you have a largest max value (>1300) then adjust your last class to that largest value in your data.

Alternatively, you can set those values to NA if you think they are outside of the valid range of NBR (in this case they are not).

You can export the rasters if you want.

```
writeRaster(x = nbr_classified,
            filename="data/week6/outputs/nbr_classified.tif",
            format = "GTiff", # save as a tif
            datatype='INT2S', # save as a INTEGER rather than a float
            overwrite = T)

writeRaster(x = landsat_nbr,
            filename="data/week6/outputs/landsat_nbr",
            format = "GTiff", # save as a tif
            datatype='INT2S', # save as a INTEGER rather than a float
            overwrite = T)
```

Your classified map should look something like:

Compare to fire boundary

As an example to see how our fire boundary relates to the boundary that we've identified using MODIS data, we can create a map with both layers. I'm using the shapefile in the folder:

`data/week6/vector_layers/fire-boundary-geomac/co_cold_springs_20160711_2200_dd83.shp`

Add fire boundary to map.

Make it look a bit nicer using a colobrewer palette. I used the `RdYlGn` palette:

`brewer.pal(5, 'RdYlGn')`

I also did a bit of legend trickery to get a box with a fill. There's probably a better way to do this!

```
legend(nbr_classified@extent@xmax-100, nbr_classified@extent@ymax,
       c("Enhanced Regrowth", "Unburned", "Low Severity", "Moderate Severity", "High Severity", "Fire bo
       col=c(rev(the_colors), "black"),
       pch=c(15,15, 15, 15, 15,NA),
       lty = c(NA, NA, NA, NA, NA, 1),
       cex=.8,
       bty="n",
       pt.cex=c(1.75))
legend(nbr_classified@extent@xmax-100, nbr_classified@extent@ymax,
       c("Enhanced Regrowth", "Unburned", "Low Severity", "Moderate Severity", "High Severity", "Fire bo
       col=c("black"),
       pch=c(22, 22, 22, 22, 22, NA),
       lty = c(NA, NA, NA, NA, NA, 1),
       cex=.8,
       bty="n",
       pt.cex=c(1.75))
```

Note that you will have to figure out what date these data are for! I purposefully didn't include it in the title of this map.

Additional Resources

- USGS Remote sensing phenology
- NASA Earth Observatory - Vegetation indices