Clouds, shadows & cloud masks in R

Learning Objectives

After completing this tutorial, you will be able to:

- Describe the impacts that thick cloud cover can have on analysis of remote sensing data.
- Use a cloud mask to remove portions of an spectral dataset (image) that is covered by clouds / shadows.
- Define cloud mask / describe how a cloud mask can be useful when working with remote sensing data.

What you need

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

Download Week 6/7 Data (~500 MB){:data-proofer-ignore=" .btn }

About Landsat scenes

The landsat satellites orbit the earth continuously collected images of the Earth's surface. The collected images, are divided into smaller regions - known as scenes.

> Landsat images are usually divided into scenes for easy downloading. Each Landsat scene is about 115 miles long and 115 miles wide (or 100 nautical miles long and 100 nautical miles wide, or 185 kilometers long and 185 kilometers wide). -*wikipedia*

In the previous lessons, we learned how to import a set of geotiffs that made up the bands of a landsat raster. However, we ran into some challenges. Our pre fire image contained a large cloud and associated shadow that covered our study area of interest - the burn scar.

Clouds and atmospheric conditions can present a significant challenge when working with spectral remote sensing data. Extreme cloud cover and shadows can render the data in those areas, un-usable given reflectance values are either washed out (too bright - as the clouds scatter all light back to the sensor) or are too dark (shadows which represent blocked or absorbed light).

In this lesson we will learn how to deal with clouds in our remote sensing data.

Let's begin by loading our spatial libraries.

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

Next, we will load the landsat bands that we loaded previously in our homework.

```r
all_landsat_bands <- list.files("data/week6/Landsat/LC80340322016189-SC20170128091153/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_st <- stack(all_landsat_bands)
```

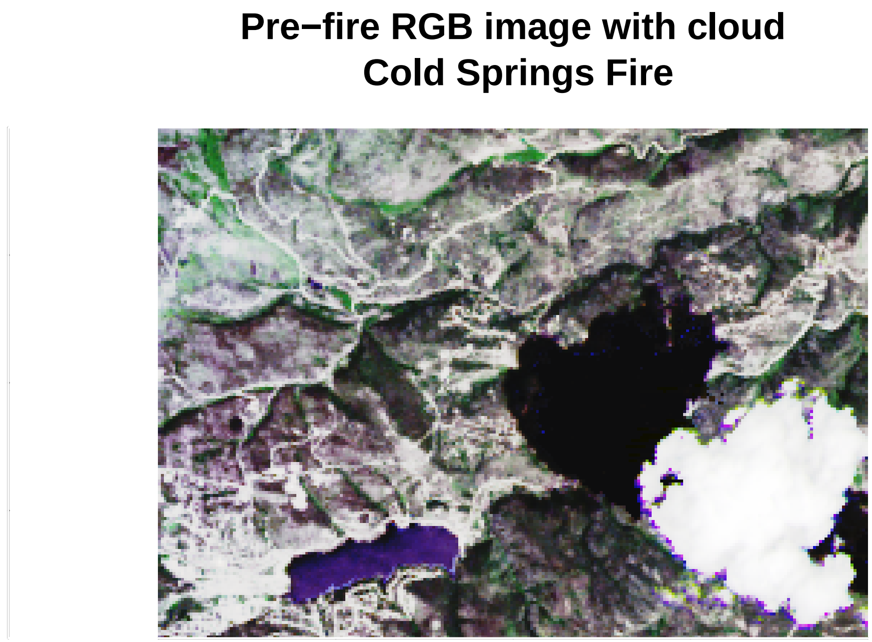

Figure 1: RGB image of our landsat data.

When we plotted the pre-fire image, we noticed there is a large cloud in our scene. Notice as i'm plotting below, i added a few *par*ameters to force R to add a title to my plot. Namely:

```
# turn the axis color to white and turn off ticks
par(col.axis="white", col.lab="white", tck=0)
# plot the data - be sure to turn AXES to T (we just color them white)
plotRGB(all_landsat_bands_st,
        r=4, g=3, b=2,
        stretch="hist",
        main="Pre-fire RGB image with cloud\n Cold Springs Fire",
        axes=T)
# turn the box to white so there is no border on our plot
box(col="white")
```

Raster masks

Often (but not always) remote sensing data come with mask layers. These layers identify pixels that are likely representative of a cloud or shadow that have been generated by whomever processed the data. When we download Landsat 8 data from Earth Explorer, the data come with 2 processed cloud mask raster layers.

1. LC80340322016189-SC20170128091153/crop/LC80340322016189LGN00_cfmask_crop.tif
2. LC80340322016189-SC20170128091153/crop/LC80340322016189LGN00_cfmask_conf_crop.tif

Let's have a look at these layers next.

```
cloud_mask_189_conf <- raster("data/week6/Landsat/LC80340322016189-SC20170128091153/crop/LC8034032201618
plot(cloud_mask_189_conf)
```

Next, we can plot the second mask layer. Do you notice any difference between the two?

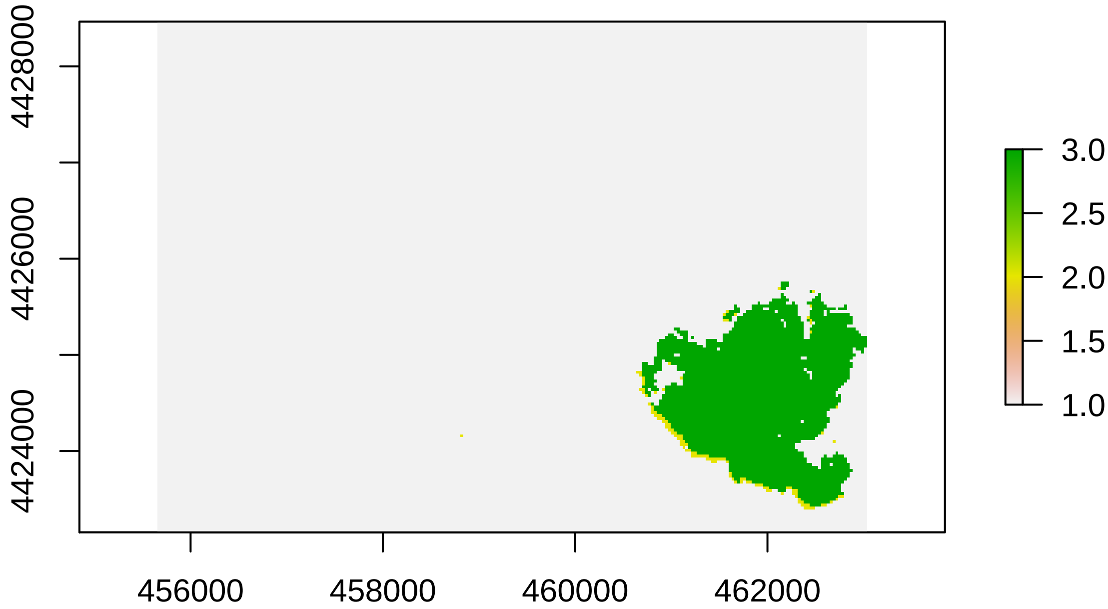

Figure 2: cloud mask - no shadows.

```
# apply shadow mask
cloud_mask_189 <- raster("data/week6/Landsat/LC80340322016189-SC20170128091153/crop/LC80340322016189LGN0
plot(cloud_mask_189)
```

What do the metadata tell us?

We just explored two layers that potentially have information about cloud cover. However what do the values stored in those rasters mean? We can refer to our metadata to learn more about how each layer in our landsat dataset are both stored and calculated.

Let's open the metadata file: `data/week6/landsat/LC80340322016189-SC20170128091153/LC80340322016189LGN00.xml` What does it tell us?

<a href="~/Documents/Github/earthlab.github.io/images/course-materials/earth-analytics/week-7/cloud-mask
<img src="~/Documents/Github/earthlab.github.io/images/course-materials/earth-analytics/week-7/cloud-mas

<figcaption>Metadata
</figcaption>

If we study the metadata we can see..

MORE HERE ON METADATA

We can use the cloud mask layer to identify pixels that are likely to be clouds or shadows. We can then set those pixel values to NA so they are not included in our quantitative analysis in R.

<a href="~/Documents/Github/earthlab.github.io/images/course-materials/earth-analytics/week-7/raster_mas
<img src="~/Documents/Github/earthlab.github.io/images/course-materials/earth-analytics/week-7/raster_ma

<figcaption>Raster masks
</figcaption>

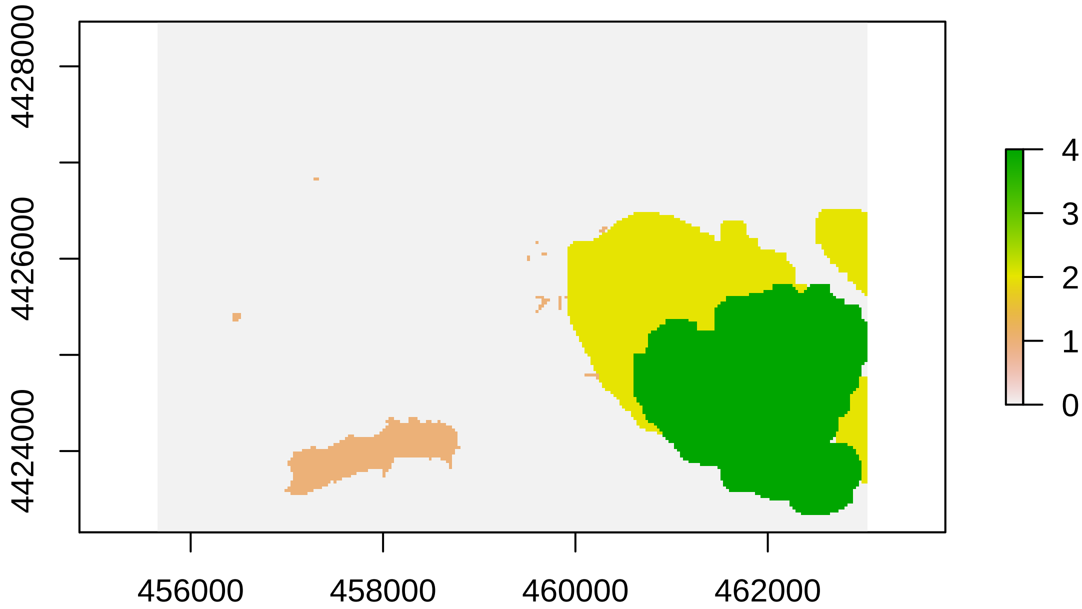

Figure 3: cloud mask with shadows

To do this we do the following.

1. we create a raster layer that is the SAME EXTENT and the same pixel resolution as our landsat scene.
2. We then set all of the values in that layer that are clouds and / or shadows to `NA`
3. Finally we use the mask() function to remove all pixels that were flagged as clouds or shadows in our mask to NA.

In this case, we want to set all values greater than 0 in the raster mask to NA.

```
# Create cloud & cloud shadow mask
cloud_mask_189[cloud_mask_189 > 0] <- NA
plot(cloud_mask_189,
     main="Our new raster mask",
     col=c("green"),
     legend=F,
     axes=F,
     box=F)

legend("topright",
       c("Not masked", "Masked"),
       fill=c("green", "white"))
```

Apply a mask

We can apply a mask to all of the bands in our raster stack which is convenient! Let's use the mask() function to mask our data.

```
# mask the stack
all_landsat_bands_mask <- mask(all_landsat_bands_st, mask = cloud_mask_189)
# plot RGB image
# first turn all axes to the color white and turn off ticks
par(col.axis="white", col.lab="white", tck=0)
```

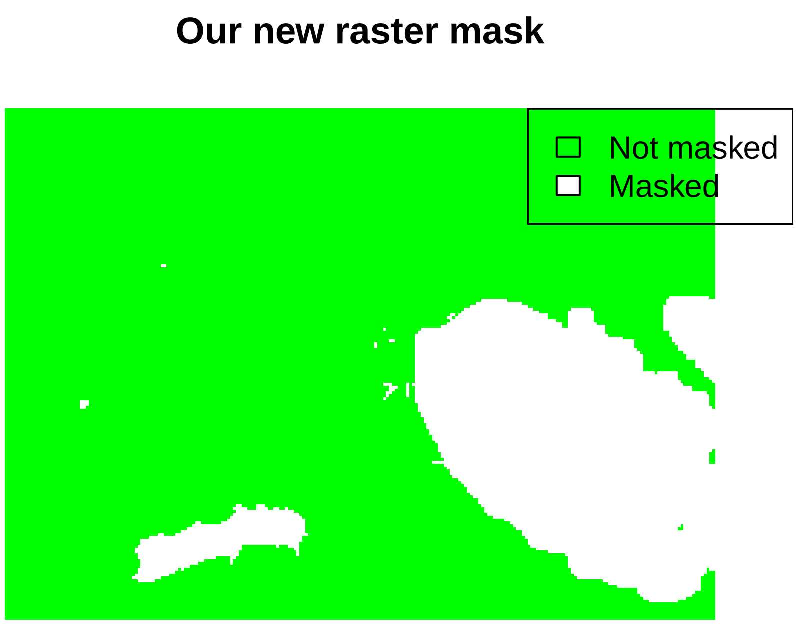

Figure 4: raster mask. green values are not masked.

```r
# then plot the data
plotRGB(all_landsat_bands_mask,
        r=4, g=3, b=2,
        stretch="hist",
        main="RGB image - are all of the clouds gone from our image?",
        axes=F)
box(col="white")
```

```r
## Create a function to calculate a veg index
get_veg_index <- function(band1, band2){
  # this function calculates the normalize difference between two bands
  # output: a new raster with index values bewteen -1 and 1
  new_index <- (band2 - band1) / (band2 + band1)
  return(new_index)
}

# calculate NBR

landsat_nbr <- overlay(all_landsat_bands_mask[[4]], all_landsat_bands_mask[[5]],
                       fun=get_veg_index)
plot(landsat_nbr)
```

Optional challenge

- overlay the fire boundary on top of the landsat pre-fire image.
- If you were asked to QUANTIFY the pre vs post fire burn area extent, what are some problems that you can anticipate running into with the cloud cover?

Figure 5: apply raster mask to stack and plot.

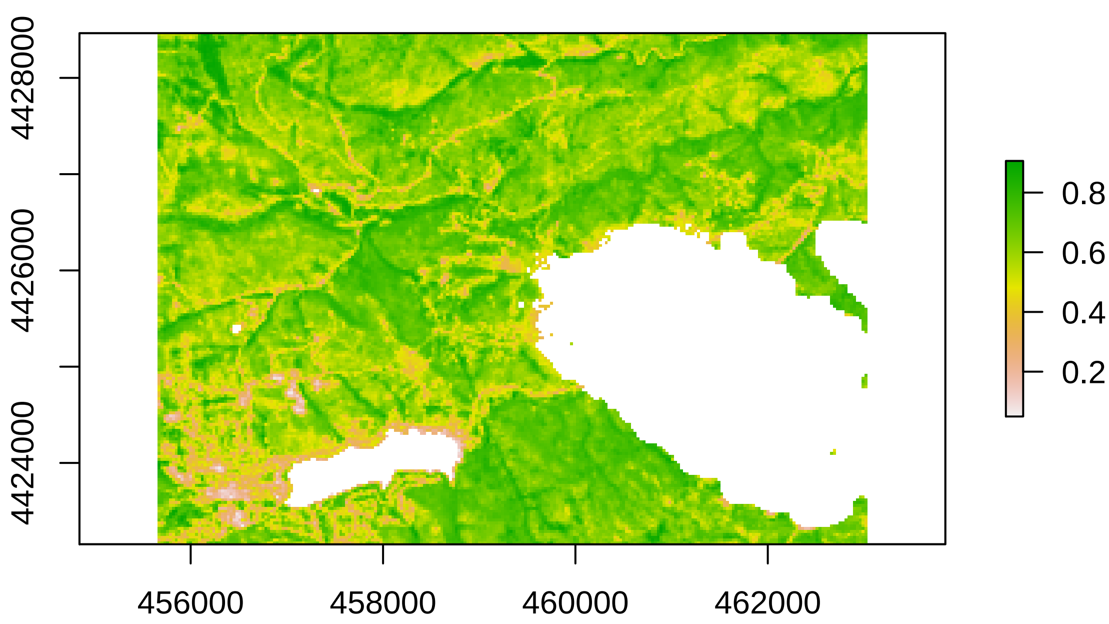

Figure 6:

good opportunity to learn about projections

let them plot modis, let them calculate NBR The overlay the fire boundary – they will have to

1. figure out band combinations
2. reproject