Classify a raster in R.

Learning Objectives

After completing this tutorial, you will be able to:

- Reclassify a raster dataset in R using a set of defined values.

What you need

You need `R` and `RStudio` to complete this tutorial. Also you should have an `earth-analytics` directory setup on your computer with a `/data` directory with it.

- How to Setup R / RStudio
- Setup your working directory
- Intro to the R & RStudio Interface

R Libraries to Install:

- **raster:** `install.packages("raster")`
- **rgdal:** `install.packages("rgdal")`

If you have not already downloaded the week 3 data, please do so now. Download Week 3 Data (~250 MB){:data-proofer-ignore=" .btn }

In this lesson, we will learn how to reclassify a raster dataset in `R`. Previously, we plotted a raster value using break points - that is to say, we colored particular ranges of raster values using a defined set of values that we call `breaks`. In this lesson, we will learn how to reclassify a raster - to create a new raster object / file that we can share with colleagues and / or open in other tools such as QGIS.

When you reclassify a raster. You assign each cell a discrete value using a defined range of values. For example above you can see that all cells that contains the values 1-3 are assigned the new value of 5. Source: ESRI.

Load libraries

```
# load the raster and rgdal libraries
library(raster)
library(rgdal)
```

Map raster values to new values

The first thing that we will need to create is a reclassification matrix. This matrix MAPS a range of values to a new defined value. Let's create a classified canopy height model where we designate short, medium and tall trees.

The values will be defined as follows:

- short = 1
- medium = 2
- tall = 3

```
# create classification matrix
reclass_df <- c(0, 10, 1,
              10, 20, 2,
              20, 35, 3)
reclass_df
## [1]  0 10  1 10 20  2 20 35  3

# reshape the object into a matrix with columns and rows
reclass_m <- matrix(reclass_df,
                 ncol=3,
                 byrow=TRUE)
reclass_m
##      [,1] [,2] [,3]
## [1,]    0   10    1
## [2,]   10   20    2
## [3,]   20   35    3
```

Reclassify raster

Next, we will reclassify the raster

```
# open canopy height model
lidar_chm <- raster("data/week3/BLDR_LeeHill/outputs/lidar_chm.tif")

# reclassify the raster using the reclass object - reclass_m
chm_classified <- reclassify(lidar_chm,
                     reclass_m)
# plot reclassified data
plot(chm_classified,
     col=c("red", "blue", "green"))
```

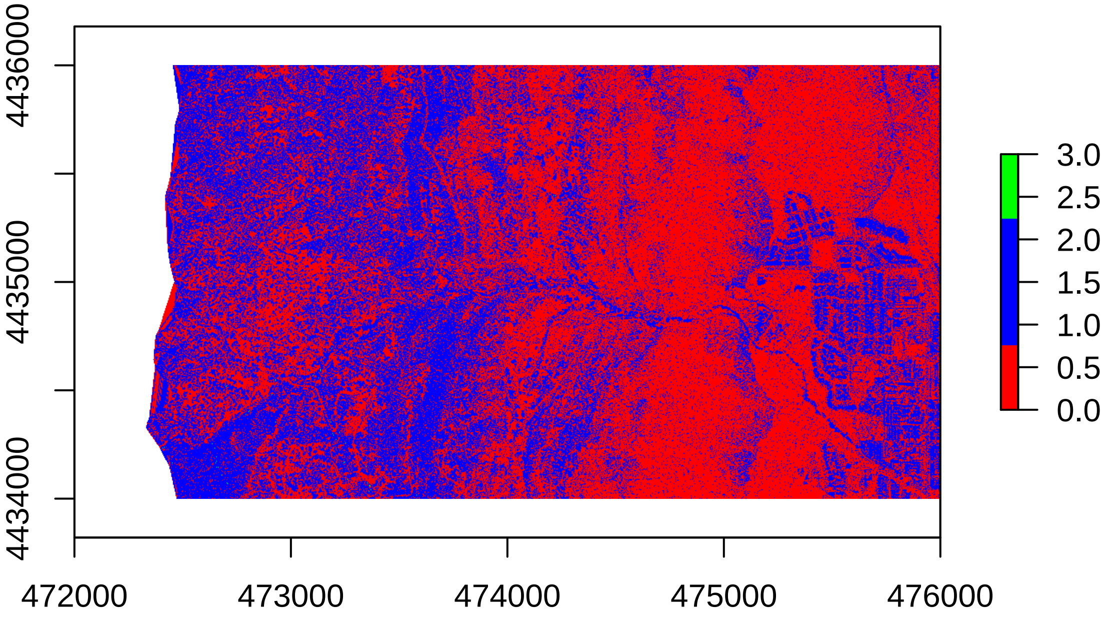

Figure 1: classified chm plot

Add custom legend

Finally, let's clean up our plot legend. Given we have discrete values we will create a CUSTOM legend with the 3 categories that we created in our classification matrix.

```r
# plot reclassified data
plot(chm_classified,
     legend=F,
     col=c("red", "blue", "green"), axes=F,
     main="Canopy Height Model - Classified: short, medium, tall trees")

legend("topright",
       legend = c("short trees", "medium trees", "tall trees"),
       fill = c("red", "blue", "green"),
       border = F,
       bty="n") # turn off legend border
```

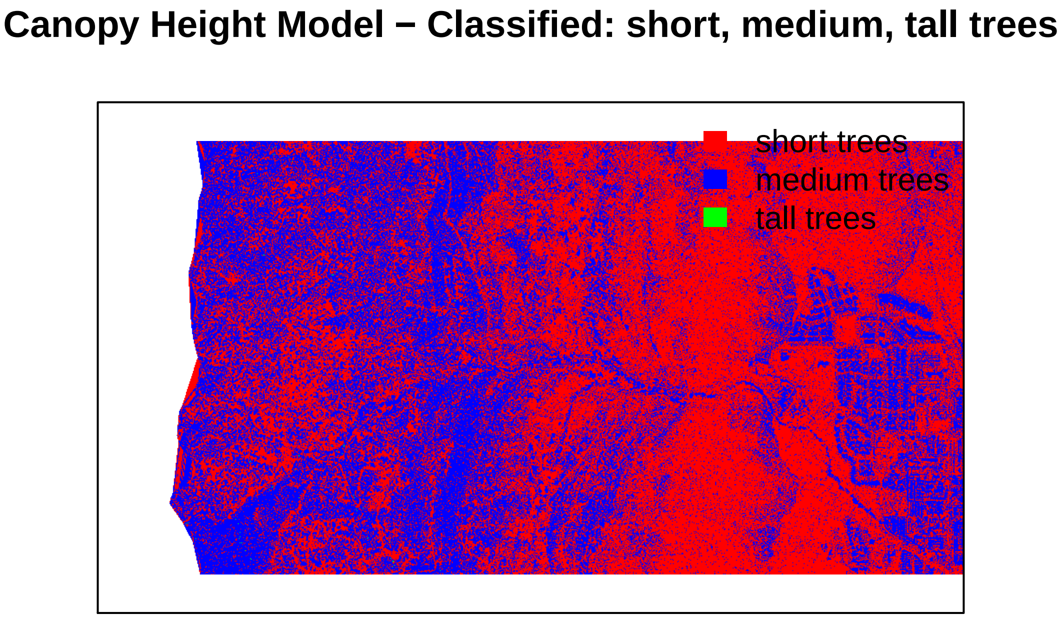

Figure 2: classified chm with legend.

Challenge - plot change over time

1. Create a classified raster map that shows **positive and negative change** in the canopy height model before and after the flood. To do this you will need to calculate the difference between two canopy height models.
2. Create a classified raster map that shows **positive and negative change** in terrain extracted from the pre and post flood Digital Terrain Models before and after the flood.

For each plot, be sure to:

- add a legend that clearly shows what each color in your classified raster represents
- use better colors that I used in my example above!.
- add a title to your plot.

You will include these plots in your final report due next week.

Check out this cheatsheet for more on colors in `R`.

Or type: `grDevices::colors()` into the r console for a nice list of colors!