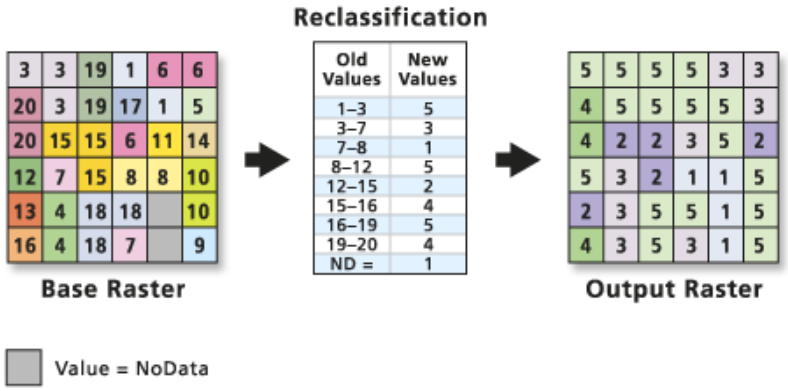
Lab 3: Working with raster data

Reclassifying Raster Data

In this lab, you will learn how to reclassify a raster dataset in Python. When you reclassify a raster, you create a new raster object / file that can be exported and shared with colleagues and / or open in other tools such as QGIS. In that raster each pixel is mapped to a new value based on some approach. This approach can vary depending upon your science question.



When you reclassify a raster, you create a new raster. In that raster, each cell from the old raster is mapped to the new raster. The values in the new raster are applied using a defined range of values or a raster map. For example, above you can see that all cells that contains the values 1-3 are assigned the new value of 5.

Dataset:

We will work with two raster datasets: a DTM and a DSM of the same area. By performing a subtraction, we will get the difference between them, which indicate the heights of the geographic objects (mostly trees). We will then classify the trees based on their heights.

Preparation:

Import related packages:

```
import rasterio as ro
from rasterio.plot import show
```

Load the two raster datasets:

```
dtm = ro.open("Data/pre_DTM.tif")
dsm = ro.open("Data/pre_DSM.tif")
```

Please write code to answer the following questions:

- What are the numbers of band(s) of these two raster datasets? (10 pts)
- What are the heights and widths of these two raster datasets? (10 pts)
- Plot out these two datasets using their default visualization. (10 pts)

Next, we will read the values (elevations) from these two datasets.

```
dtm_data = dtm.read(1,masked=True)
dsm_data = dsm.read(1,masked=True)
```

Note that “masked=True” will automatically remove the cells with missing values

Subtracting the two gives us the heights of the geographic features:

```
heights = dsm_data - dtm_data
```

We can find the maximum height and minimum height in our data:

```
heights.max()
heights.min()
```

Next, we will plot a histogram of the heights to see the height distribution:

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(8,8))

ax.hist(heights.ravel(), bins = range(0,31,1),color='purple', edgecolor='white')
ax.set_title("Distribution of heights",fontsize = 16)
ax.set(xlabel="Height (m)", ylabel="Number of Pixels");
```

In the next step, we will reclassify the raster cells based the heights. Particularly, we will have:

- No trees: (0m - 2m tall)
- Short trees: (2m - 7m tall)
- Medium trees: (7m - 12m tall)
- Tall trees: (> 12m tall)

We will reclassify the raster using the rule below. Notice in the matrix below we use *Inf* to represent the largest or max value found in the raster. So our assignment is as follows:

- 0 - 2 meters -> 1
- 2 - 7 meters -> 2 (short trees)
- 7 - 12 meters -> 3 (medium trees)
- > 12 or 12 - Inf -> 4 (tall trees)

np.digitize

Numpy has a function called digitize that is useful for classifying the values in an array. This function will replace each datapoint with an integer corresponding to which value range it belongs to. Perform the reclassification:

```
import numpy as np

new_class_bins = [0, 2, 7, 12, np.inf]   # define the value ranges

# reclassify
reclassified_raster = np.digitize(heights, new_class_bins)
```

Visualize the reclassified result:

```
fig, ax = plt.subplots(figsize=(8,4))
ax.imshow(reclassified_raster)
```

Use the dsm data as a template to save your reclassification result into a new raster (20 pts): Hint you will need to convert the dtype of your reclassification result using:

reclassified_raster.astype(dsm_data.dtype)

Please submit your Jupyter notebook (a full working notebook will get 50 pts; any mistake will result in 10 pts deducted):

- Lab3_FirstName_LastName.ipynb