Spatial Analysis

Oleksii Yehorchenkov

2023-11-30

This tutorail is based on Intro to spatial analysis in R (https://ourcodingclub.github.io/tutorials/spatial/) by Maude Grenier

Loading packages

```
library(tidyverse)
library(raster)
library(sf)
library(viridisLite)
library(rasterVis)
```

Load data

```
tay <- raster("./data/taycrop.tif")
```

Get properties of the Tay raster

```
tay
```

```
## class      : RasterLayer
## band       : 1  (of  12  bands)
## dimensions : 507, 848, 429936  (nrow, ncol, ncell)
## resolution : 9.217891e-05, 9.217891e-05  (x, y)
## extent     : -4.320218, -4.242051, 56.45366, 56.50039  (xmin, xmax, ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs
## source     : taycrop.tif
## names      : taycrop_1
```

We can create individual raster layers for each of the spectral bands in the raster tay.

```
b1 <- raster('./data/taycrop.tif', band=1)
b2 <- raster('./data/taycrop.tif', band=2)
b3 <- raster('./data/taycrop.tif', band=3)
b4 <- raster('./data/taycrop.tif', band=4)
b5 <- raster('./data/taycrop.tif', band=5)
b6 <- raster('./data/taycrop.tif', band=6)
b7 <- raster('./data/taycrop.tif', band=7)
b8 <- raster('./data/taycrop.tif', band=8)
b9 <- raster('./data/taycrop.tif', band=9)
b10 <- raster('./data/taycrop.tif', band=10)
b11 <- raster('./data/taycrop.tif', band=11)
b12 <- raster('./data/taycrop.tif', band=12)
```

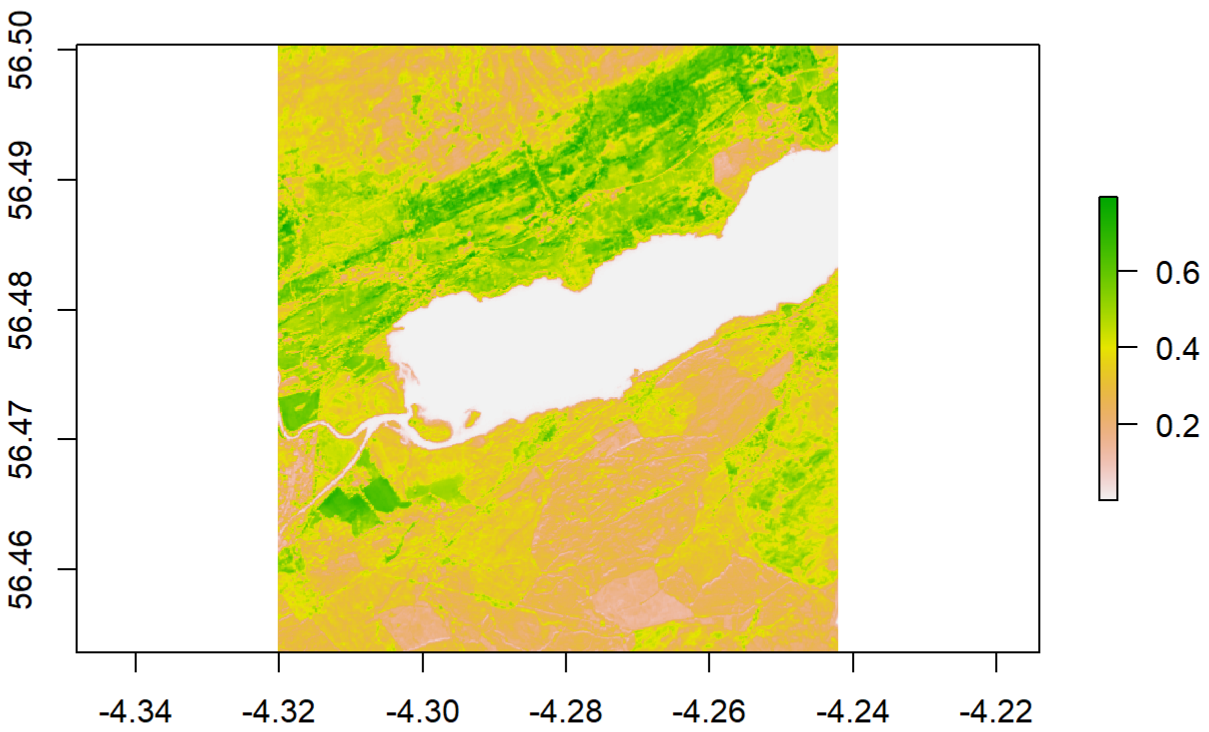
We can now compare two bands to see if they have the same extent, number of rows and column, projection, resolution and origin. As can be seen below, bands 2 and 3 match.

```
compareRaster(b2, b3)
```

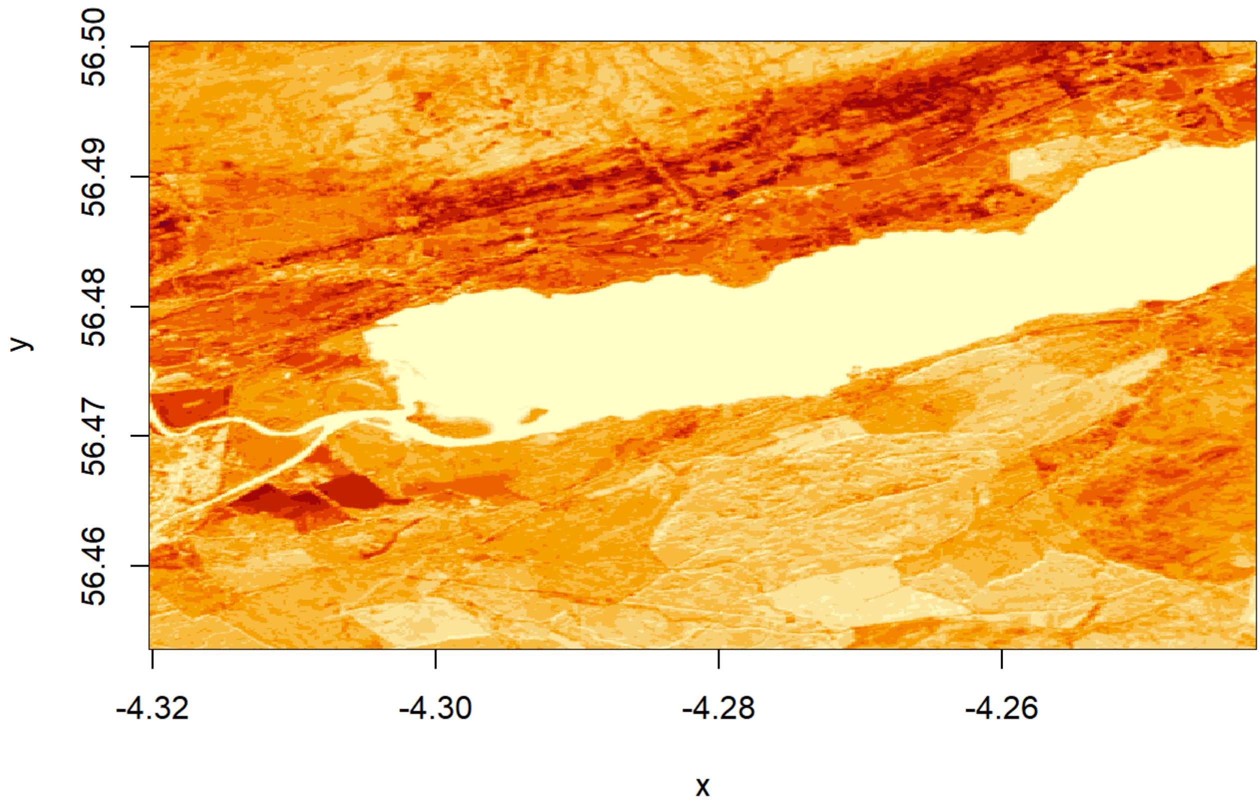
```
## [1] TRUE
```

The bands can be plotted using the plot or image function. Note that the plot function only plots 100,000 pixels but image strectches the view.

```
plot(b8)
```



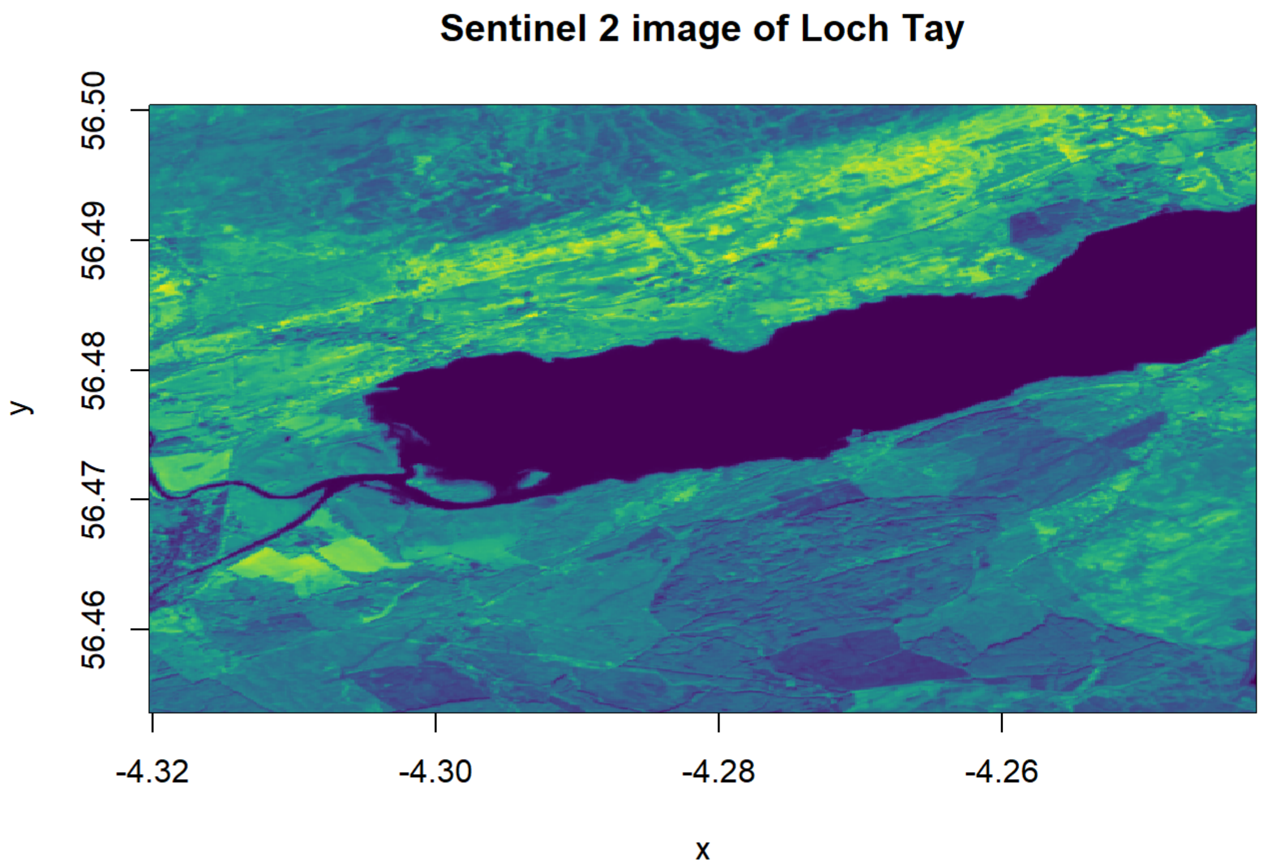

```
image(b8)
```



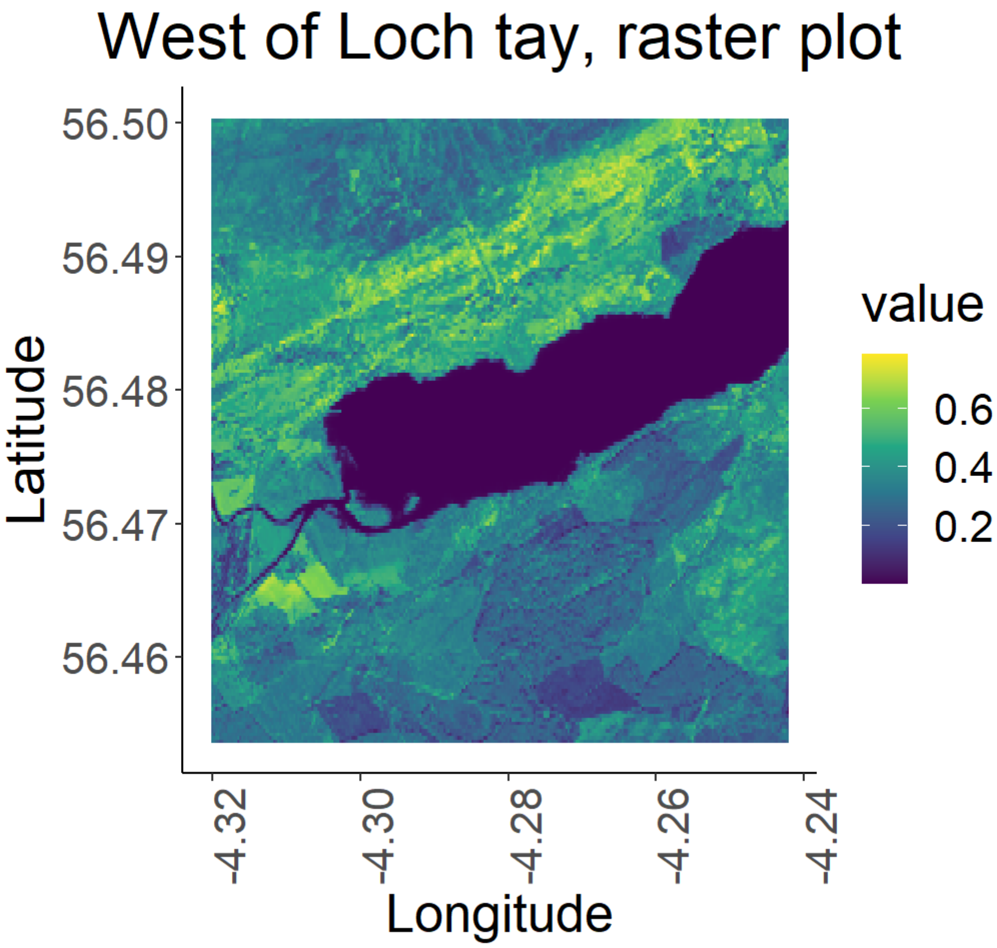
Visualise spectral bands

The bands can be plotted with different colour palettes to improve visualisation, such as viridis, and saved using the code below

```
image(b8, col=viridis(256, option="D"), main="Sentinel 2 image of Loch Tay")
```




```
gplot(b8) +
  geom_raster(aes(x = x, y = y, fill = value)) +
  # value is the specific value (of reflectance) each pixel is associated with
  scale_fill_viridis_c() +
  coord_quickmap() +
  ggtitle("West of Loch tay, raster plot") +
  xlab("Longitude") +
  ylab("Latitude") +
  theme_classic() +                                    # removes defalut grey background
  theme(plot.title = element_text(hjust = 0.5),             # centres plot title
        text = element_text(size=20),                       # font size
        axis.text.x = element_text(angle = 90, hjust = 1))  # rotates x axis text
```

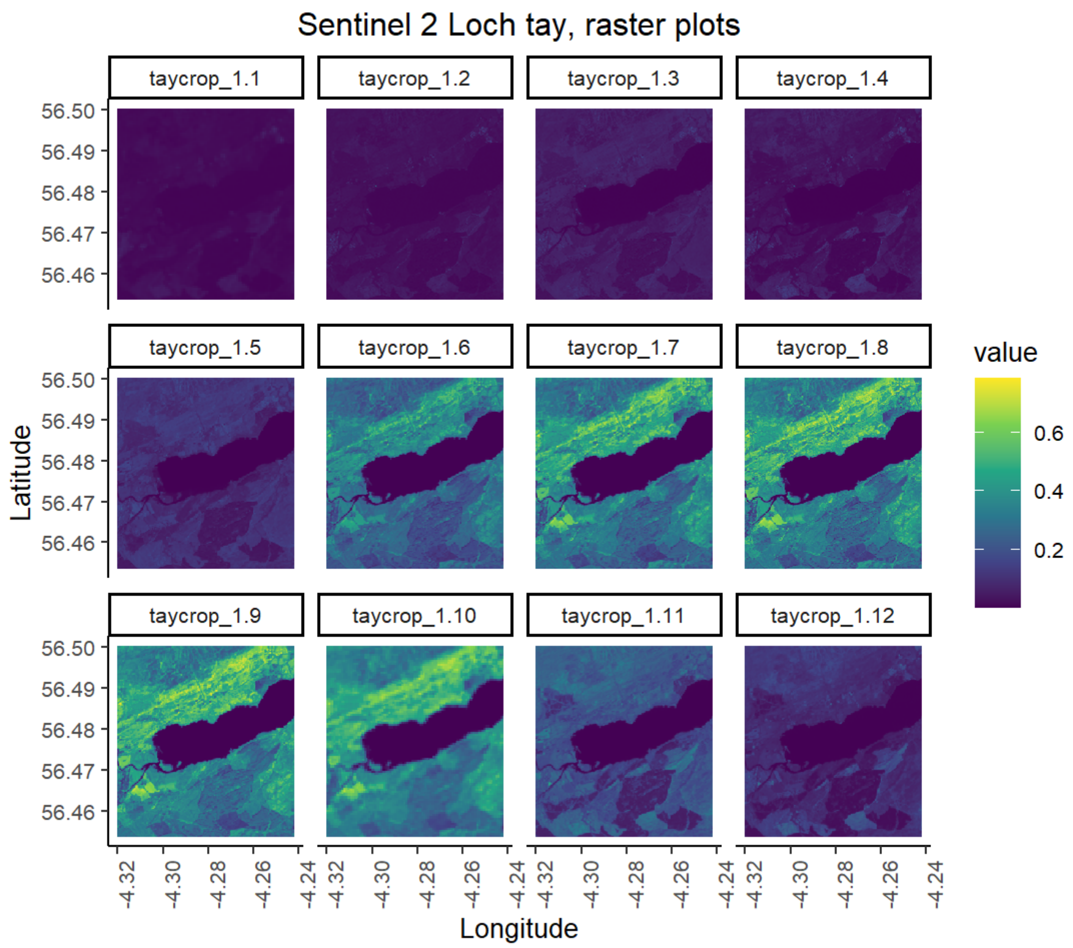


To visualise all the bands together, we can use facet_wrap in gplot. First, we will create a stack of all the bands, so just putting them all on top of each other, like layers in a cake.

```
t <- stack(b1,b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12)
```

Now we are ready to make out facetted plots.

```
gplot(t) +
  geom_raster(aes(x = x, y = y, fill = value))+
  scale_fill_viridis_c() +
  facet_wrap(~variable) +
  coord_quickmap()+
  ggtitle("Sentinel 2 Loch tay, raster plots") +
  xlab("Longitude") +
  ylab("Latitude") +
  theme_classic() +
  theme(text = element_text(size=10),
        axis.text.x = element_text(angle = 90, hjust = 1)) +
  theme(plot.title = element_text(hjust = 0.5))
```



Notice the difference in colour and range of legend between the different bands. Different earth surfaces reflect the solar radiation differently and each raster layer represents how much incident solar radiation is reflected at a particular wavelength bandwidth. Bands 6 to 9 are in the Near Infrared Range (NIR). Vegetation reflects more NIR than other wavelengths but water absorbs NIR, therefore the lighter areas with high reflectance values are likely to be vegetation and the dark blue, low reflectance value areas, likely to be water. Also note that the Sentinel 2 bands have 3 levels of spatial resolution, 10 m, 20 m, and 60 m (see summary below).

10 m resolution band 2, band 3, band 4 and band 8

20 m resolution band 5, band 6, band 7, band 11 and band 12

60 m resolution band 1, band 9 and band 10

Manipulate rasters: NDVI classification

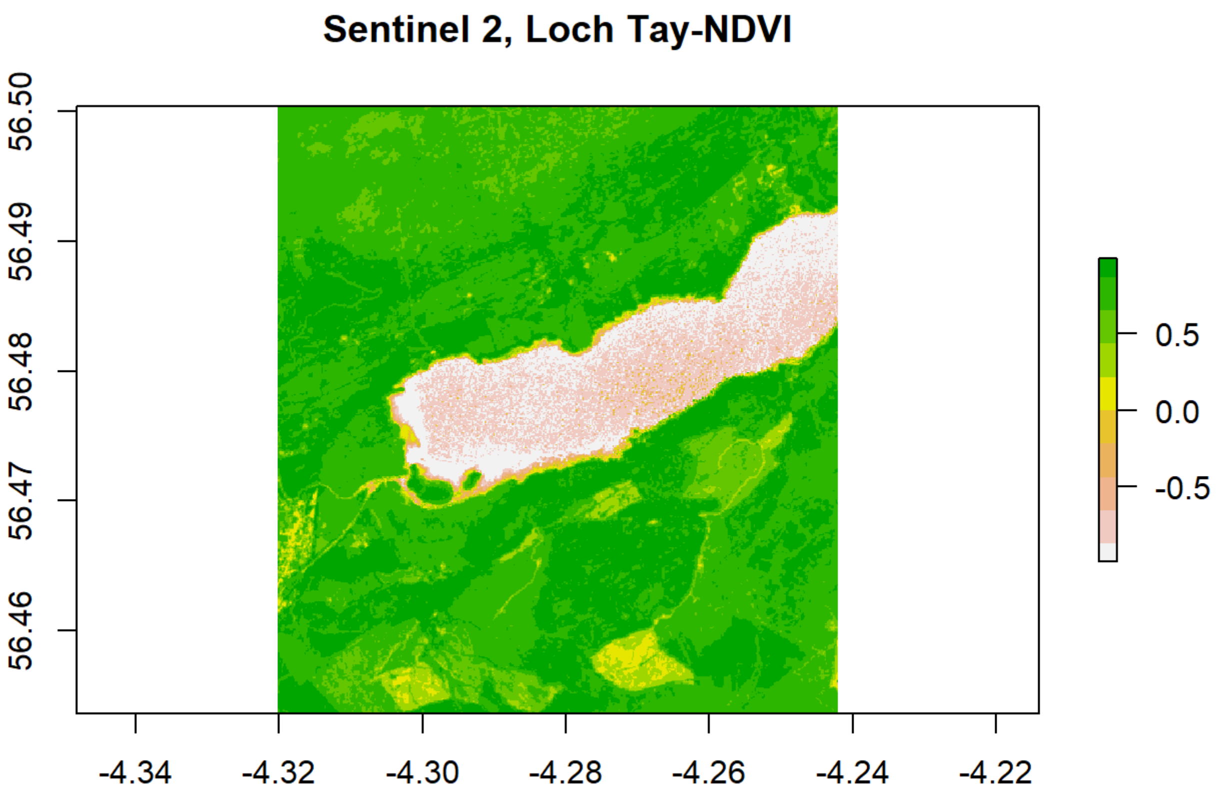
he **Normalised Difference Vegetation Index (NDVI)** is a widely used vegetation index that quantifies vegetation presence, health or structure. It is calculated using the Near Infrared (NIR) and Red bandwith of the spectrum. Healthy vegetation reflects light strongly in the NIR part of the spectrum and absorbs light in red part of the visible spectrum for photosynthesis. A high ratio between light refected in the NIR part of the spectrum and light reflected in the red part of the spectrum would represent areas that potentially have healthy vegetation. It is worth noting that different plant species absorb light in the red part of the spectrum at different rates. The same plant will also absorb light in the red band differently depending on whether it is stressed or healthy, or the time of year. It is often used over large areas as an indication of land cover change.

The **NDVI** ratio is calculated using `(NIR - Red) / (NIR + Red)`. For example, a pixel with an **NDVI** of less than 0.2 is not likely to be dominated by vegetation, and an **NDVI** of 0.6 and above is likely to be dense vegetation.

In R, we can calculate the **NDVI** by creating a function and using raster math operations where `NIR = band 8` and `Red = band 4` in Sentinel 2 images. We will first use the raster brick we created earlier from the original file.

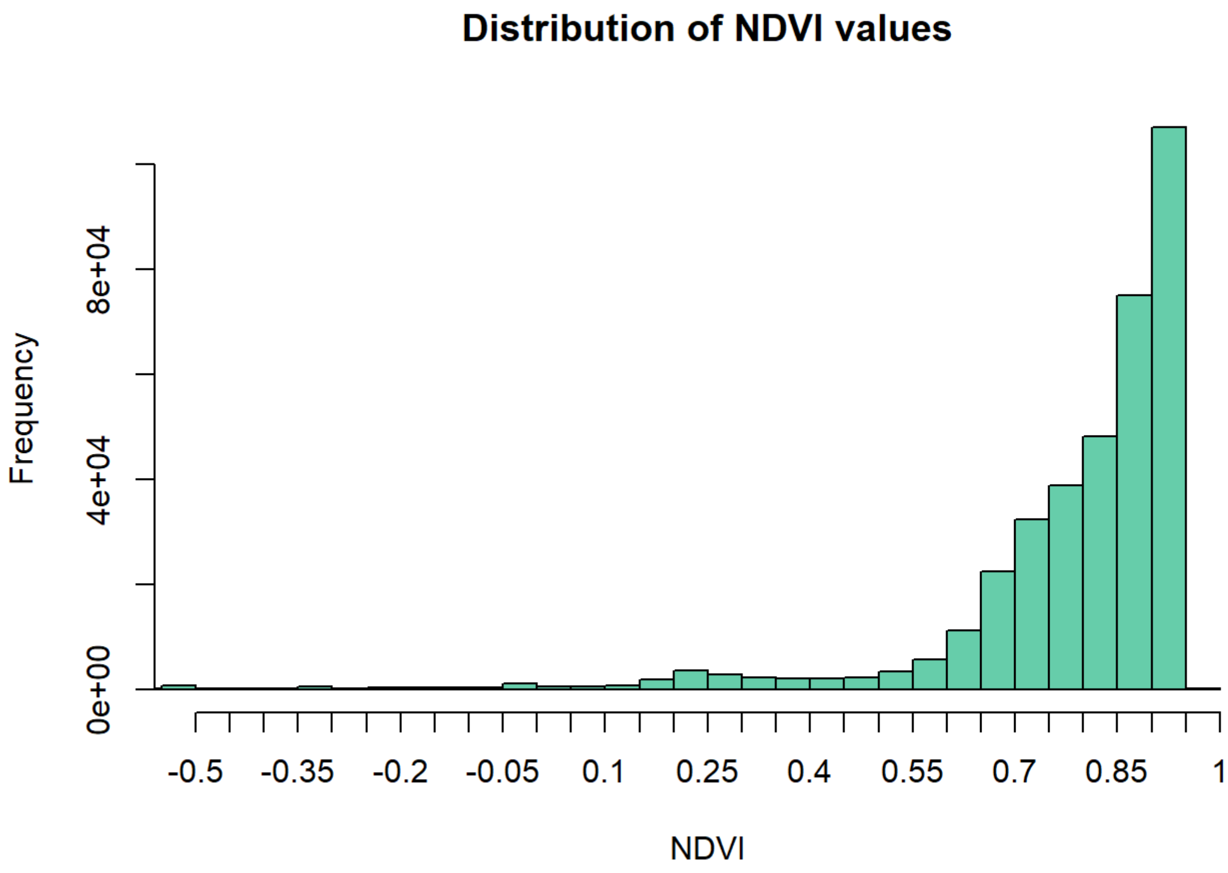
```
# NDVI

# Created a VI function (vegetation index)
VI <- function(img, k, i) {
  bk <- img[[k]]
  bi <- img[[i]]
  vi <- (bk - bi) / (bk + bi)
  return(vi)
}

# For Sentinel 2, the relevant bands to use are:
# NIR = 8, red = 4
```

```
s_tay <- brick('./data/taycrop.tif') # load original file as a raster brick

ndvi <- VI(s_tay, 8, 4)
# 8 and 4 refer to the bands we'll use

plot(ndvi, col = rev(terrain.colors(10)), main = 'Sentinel 2, Loch Tay-NDVI')
```



To find out the distribution of the pixel NDVI values, we can plot a histogram.

```
# Create histogram of NDVI data
hist(ndvi,
     main = "Distribution of NDVI values",
     xlab = "NDVI",
     ylab= "Frequency",
     col = "aquamarine3",
     xlim = c(-0.5, 1),
     breaks = 30,
     xaxt = 'n')
axis(side = 1, at = seq(-0.5,1, 0.05), labels = seq(-0.5,1, 0.05))
```



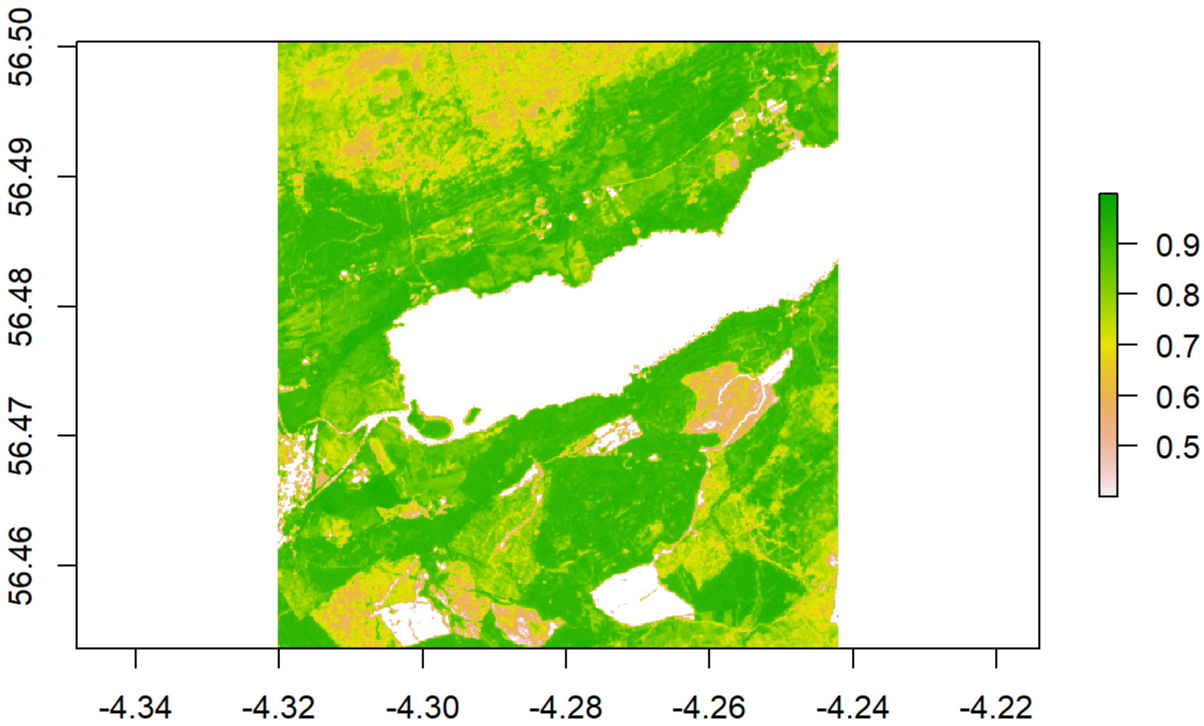
So what does this mean?

The histogram is strongly skewed to the right, towards highh NDVI values, indicating a highly vegetated area.

Now that we know that this area has lots of vegetation, we can also mask the pixels with an NDVI value of less than 0.4 (less likely to be vegetation) to highlight where the vegetated areas occur.

```
# Mask cells that have NDVI of less than 0.4 (less likely to be vegetation)

veg <- reclassify(ndvi, cbind(-Inf, 0.4, NA))
# We are reclassifying our object and making all values between
# negative infinity and 0.4 be NAs

plot(veg, main = 'Veg cover')
```

Veg cover


56.50
56.49
56.48
56.47
56.46
-4.34
-4.32
-4.30
-4.28
-4.26
-4.24
-4.22
0.9
0.8
0.7
0.6
0.5