Exploring the variability of snow in Iceland

focusing on Fagradalsfjall area

Carlotta Zanetti

Alma Mater Studiorum UniBO

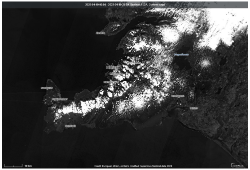
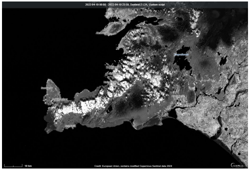
Link to my GitHub account:
`https://github.com/carlottazanetti`

Outline

Sentinel 2 dataset

Why Sentinel 2 dataset?

- Easy to download from the Copernicus website `https://dataspace.copernicus.eu/browser/`
- Good resolution
- Possibility to look at different channels directly from the website

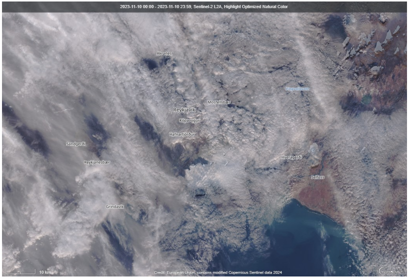




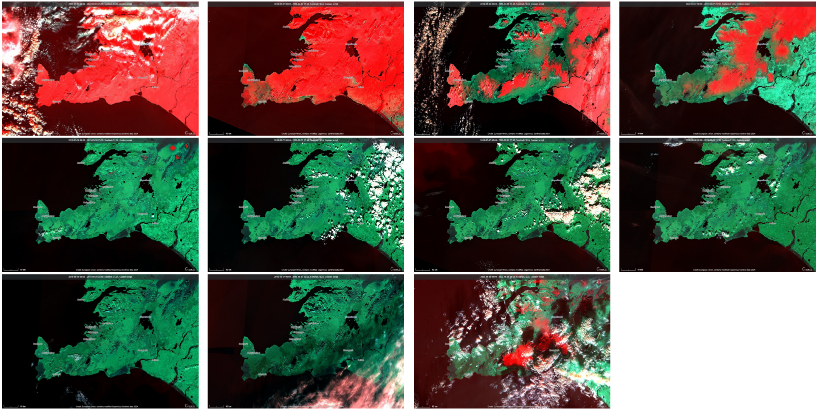
Outline

Importing images

For each month an image was picked to be representative of the snow cover.

The images come from different years since the presence of clouds didn't allow to collect all the data from the same year.
Also no data was available for December.



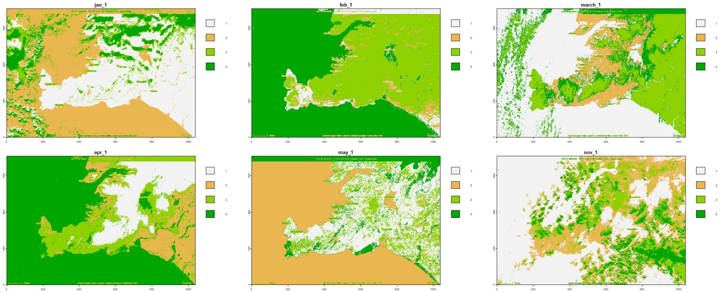


In this case I used an RGB with bands 0.4, 1.6, 2.1 microns to differentiate snow (red) from water clouds (white)

Snow detection

After importing the images with the function rast(), im.classify() from package imageRy was used to classify the images in four levels of energy.

```
setwd('C:/Users/carlo/Desktop/MONITORING ECOSYSTEMS/exam')
jan <- rast('jan.jpg')
jan_cluster <- im.classify(jan, 4)
```



Snow cover

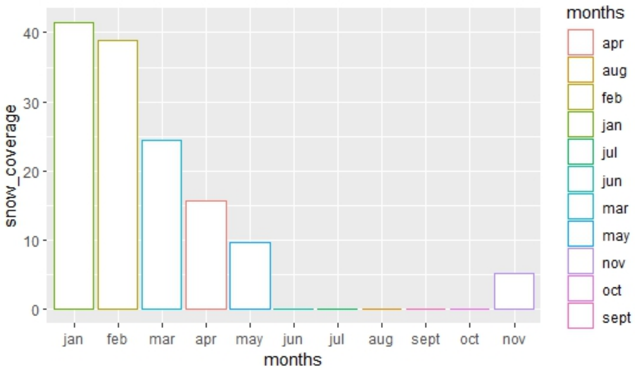
For each image with some snow cover, the level corresponding to snow was detected and the percentage of pixels were calculated

```
freq_jan <- freq(jan_cluster[[1]])
snow_jan <- freq_jan[[3]][[1]]
tot_jan <- ncell(jan_cluster[[1]])
percentage_jan <- snow_jan * 100 / tot_jan
percentage_jan
```

Snow cover

The results were plotted in a histogram

```
ggplot(results, aes(x=months, y=snow_coverage, color=months))
+ geom_bar(stat="identity", fill="white") +
scale_x_discrete(limits = months)
```

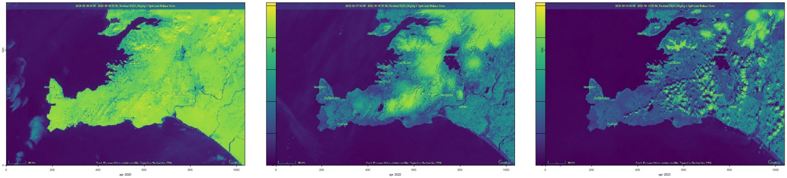


Outline

Importing images

In this case, pictures from April 2020, 2022, 2023 were taken.

```
snow <- c(apr2020[[1]], apr2022[[1]], apr2023[[1]])
#Plotting the data
cl_vir <- colorRampPalette(viridis(7))(255)
par(mfrow = c(1,3))
plot(snow[[1]], col=cl_vir, xlab='apr 2020')
plot(snow[[2]], col=cl_vir, xlab= 'apr 2022')
plot(snow[[3]], col=cl_vir, xlab= 'apr 2023')
```

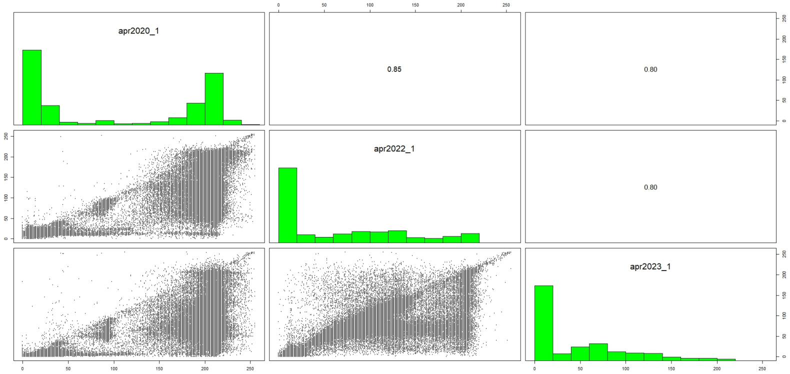


Correlation

```
pairs(snow)
```

Using the function pairs() to investigate the correlation between different years. Let's analyse the reflectances:

- Pixels with low reflectance represent the ocean
- Pixels with high reflectance represent snow



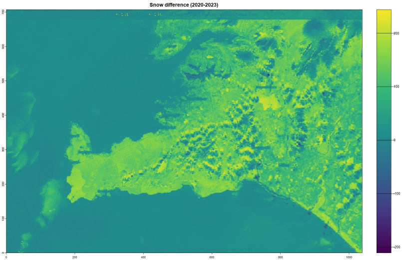
Snow differences

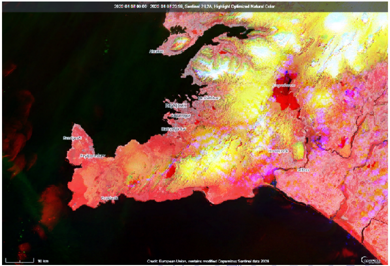
To highlight the variability between the year 2020 and 2023, their difference was plotted

```
diff= snow[[1]] - snow[[3]]
plot(diff, col=cl_vir)
```

An RGB with 2020 in the red channel, 2022 in the green channel and 2023 in the blue channel shows where the contribution of each year is stronger

```
im.plotRGB(snow, r=1, g=2, b=3)
```





Outline

False color RGB

To analyse the Fagradalsfjall volcanic eruption occurred in 2010, I decided to create an RGB using bands 2.1, 0.8 and 0.6 microns. This false color RGB is used to spot volcanic eruptions and fires:

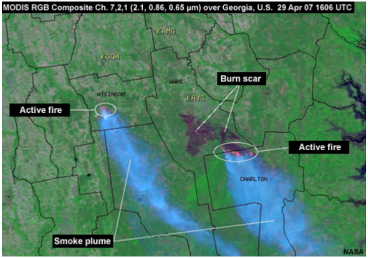
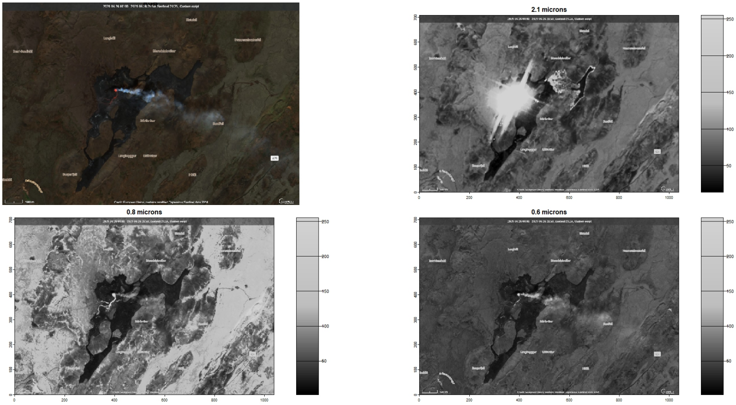


Figure: example of a fire over Georgia spotted with Modis

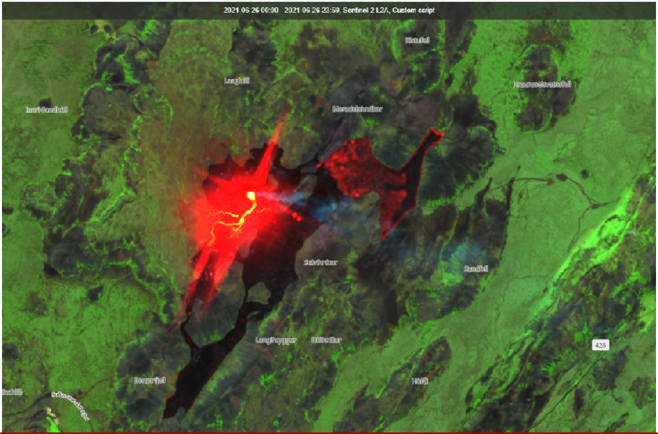
- Lava (or fire) will appear red due to the strong reflectivity in the 2.1 band
- Vegetation will appear green due to the strong reflectivity in the 0.8 channel, hence burnt areas will appear brown
- The smoke plume wil appear cyan, due to the high reflectivity in the visible band

Importing images



Creating the RGB

```
#Stacking images all together
FalseRGB <- c(b2.1[[1]], b0.8[[1]], b0.6[[1]])
#Plotting the image in false RBG colors
plotRGB(FalseRGB, r=1, g=2, b=3)
```



Standard deviation

How to decide which band to pick to calculate the standard deviation? We can perform the Principal Component Analysis to retrieve the pc1, which will be the layer that explains variability the most.

```
#Performing PCA on FalseRGB
pc <-  im.pca2(FalseRGB)
```

```
Standard deviations (1, .., p=3):
[1] 38.53183 28.53461 14.15930

Rotation (n x k) = (3 x 3):
                           PC1        PC2         PC3
iceland_eruption_2.1_1 -0.1694865  0.9814425  0.08969322
iceland_eruption_0.8_1 -0.9707964 -0.1505799 -0.18676220
iceland_eruption_0.6_1 -0.1697904 -0.1187275  0.97830210
```

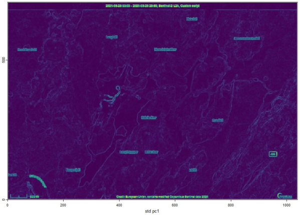
By looking at the table, it's easy to recognise which band contributes the most to pc1: it's the 0.8 band.

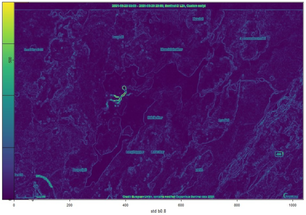
Standard deviation

The standard deviation is calculated using the function focal() combined with the std function. In this case the std is calculated on a moving window of 3x3 pixels

```
pc1 <- pc$PC1
pc1sd3 <- focal(pc1, matrix(1/9,3,3), fun=sd)
```

The std can also be calculated on the 0.8 band





As expected, the two images are similar, since band 0.8 contributes the most to pc1. Values of high std can generally be found on boundaries, indicating high geological variability.

Thank you!