IMPACT OF CLIMATE CHANGE ON MOUNTAINOUS ECOSYSTEMS

Evaluation of the Ecological Effects of Climate Change in the Alta Valtellina Area

Index

1. Hypothesis

- The research question posed to this analysis is:

How much is the vegetation of the mountainous ecosystems affected by the ongoing rise in temperature and drought stress?

2. Scope of the work

- To analyse the impact of drought on vegetation using NDVI (Sentinel-2) aggregated by plants seasonal phase.
- The period we will focus on is from 2022 to 2023, during this period, Italy experienced extreme events, including droughts and floods.

Year	Temperature Anomaly	Precipitation	Notes
2022	+1.23 °C	-22% below average	Hottest and driest year on record in Italy
2023	+1.12 °C	Near average	Second warmest year; record-high minimum temps

https://www.isprambiente.gov.it/en/archive/news-and-other-events/ispra-news/2023/07/record-heat-and-drought-in-2022
https://www.eea.europa.eu/en/analysis/indicators/global-and-european-temperatures

The objective is to **assess the impact** of the changing conditions on mountainous ecosystems in an area in Alta Valtellina of **1912.74 km^2**.

3. Plant seasonal phases considerations

- To avoid distortions caused by dormancy (very low NDVI), the analysis focuses on **four key plant developmental phases**, each lasting ~30 days.
- These intervals help track seasonal NDVI dynamics and detect drought-related anomalies in Italy between 2022 and 2023.

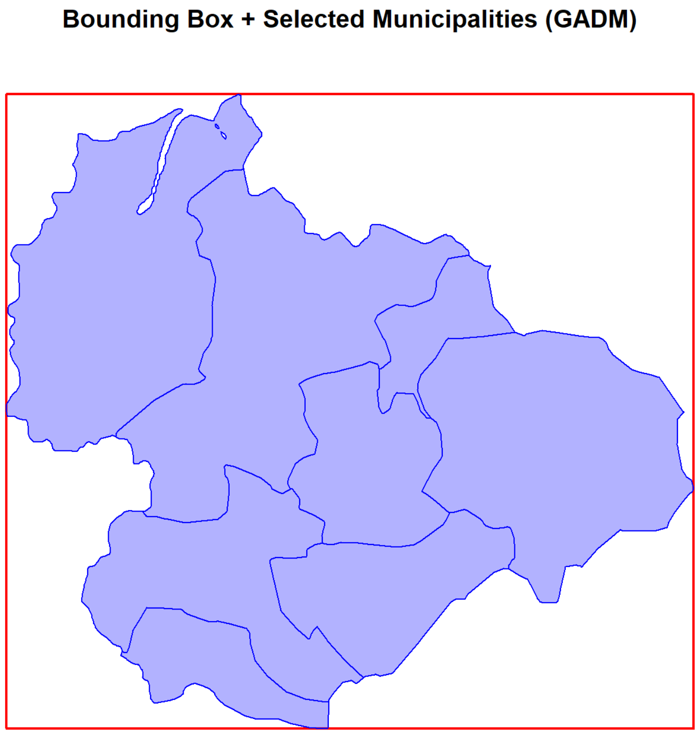
Phase	Time Period	Description
Vegetative Awakening	May 10 – June 9	First leaves appear; early flowering begins. NDVI is low but rapidly increasing.
Maximum Activity	June 25 – July 24	Peak photosynthesis and full leaf coverage. NDVI reaches its maximum.
Summer Stress	August 9 – September 9	Possible heat or drought stress. NDVI stabilizes or slightly decreases.
Early Senescence	September 25 – October 25	Leaves begin to yellow, photosynthetic activity declines. NDVI shows a clear drop.

Package used for the analysis

```
# install.packages(c("sf", "geodata", "viridis", "imageRy", "tidyr", "ggplot2", "patchwork", "raster", "terra"))



library(sf)        # For handling spatial vector data (e.g., polygons, shapefiles)
library(geodata)   # For downloading administrative boundaries (e.g., GADM dataset)
library(viridis)   # Color palettes optimized for visibility and accessibility
library(imageRy)   # Used for working with raster/image data
library(tidyr)     # For data wrangling and reshaping
library(ggplot2)   # For creating elegant data visualizations
library(patchwork) # For combining multiple ggplot2 plots
library(raster)    # For working with raster data (e.g., satellite images)
library(terra)     # Newer alternative to 'raster' for raster and vector data
```

4. Administrative boundaries and Polygon

- The bounding box used to extract Sentinel-2 data was generated by selecting **eight key mountain municipalities** in the Province of Sondrio using **GADM level-3 administrative boundaries (geodata library)**.
- After filtering and validating the spatial geometries, a rectangular bounding box was computed to enclose the selected municipalities.




```r
# Load Italy's municipal boundaries (GADM level 3) and convert to 'sf'
italy_admin3 <- st_as_sf(gadm("ITA", level = 3, path = tempdir()))

# Select target municipalities in Sondrio Province
target_municipalities <- c("Livigno", "Valdidentro", "Valdisotto", "Bormio",
                           "Valfurva", "Grosio", "Sondalo", "Grosotto")
selected <- italy_admin3[italy_admin3$NAME_3 %in% target_municipalities, ]
selected <- st_make_valid(selected)  # Fix geometries if needed

# Create bounding box polygon around selected municipalities
bbox <- st_as_sfc(st_bbox(selected))
bbox <- st_sf(geometry = bbox, crs = st_crs(selected))

# Plot bbox and municipalities
plot(st_geometry(bbox), border = "red", lwd = 2, main = "Bounding Box + Selected Municipalities")
plot(st_geometry(selected), add = TRUE, col = rgb(0, 0, 1, 0.3), border = "blue")

# Export bounding box as WKT
st_as_text(st_geometry(bbox))
```

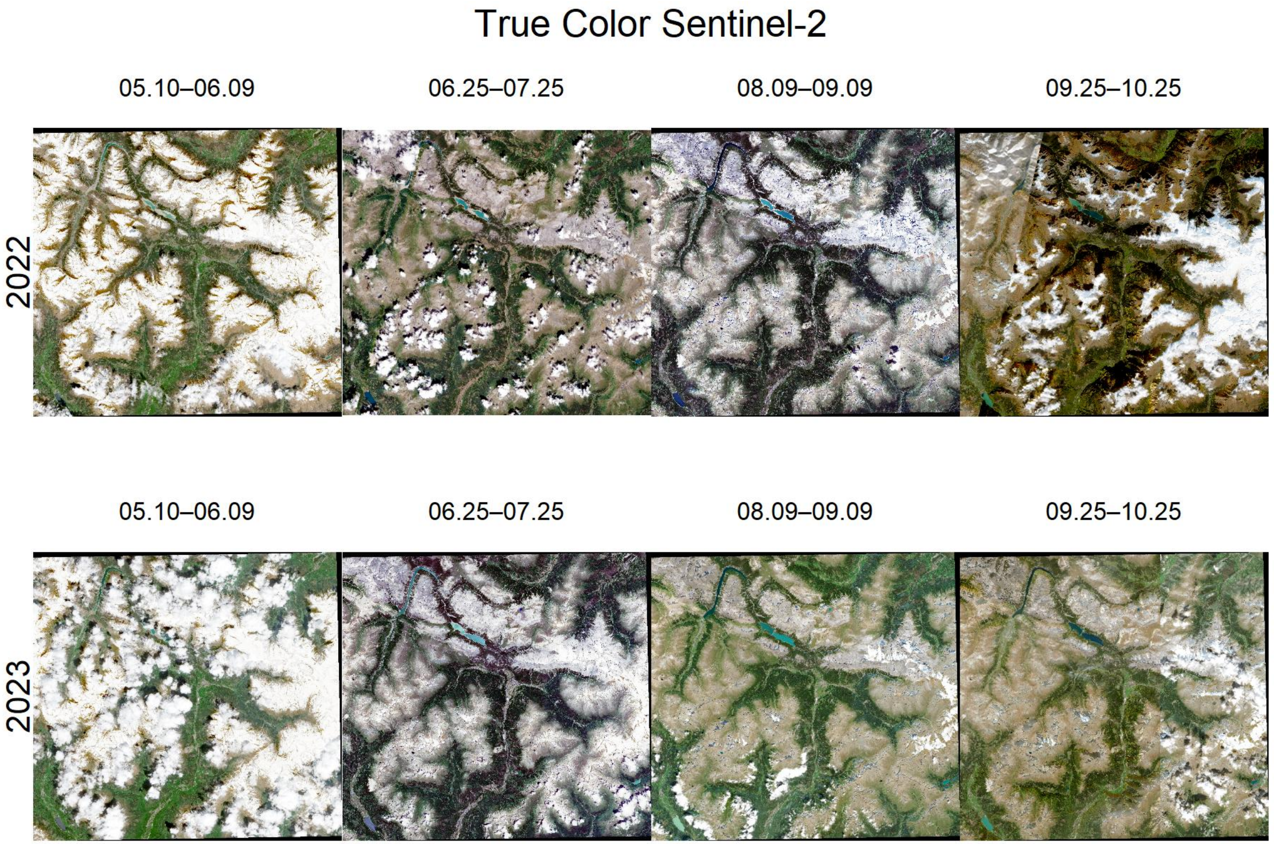
- This polygon was then converted into a **spatial feature (sf)** and exported.

```
# "POLYGON ((10.03806 46.26057, 10.63152 46.26057, 10.63152 46.63806, 10.03806 46.63806, 10.03806 46.26057))"
```

5. Visualization of the study area: TRUE COLOUR

- For the true colour it was possible to download 6 images out of the 8 periods, two images were repeatedly **corrupted**:
    - 2022-08-09 and 2023-06-25

```
# True Color Final
par(mfrow = c(2, 4), mar = c(1, 2, 3, 1), oma = c(0, 3, 3, 2))  # Margins and layout

# 2022
plotRGB(rasters_true_color[[1]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("05.10-06.09", side = 3, line = 1, cex = 1)
mtext("2022", side = 2, line = 2, cex = 1.3, las = 3)
plotRGB(rasters_true_color[[2]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("06.25-07.25", side = 3, line = 1, cex = 1)
plotRGB(true_color_2289_2299, r = 1, g = 2, b = 3, scale = 10000, stretch = "hist", maxcell = Inf)
mtext("08.09-09.09", side = 3, line = 1, cex = 1)
plotRGB(rasters_true_color[[4]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("09.25-10.25", side = 3, line = 1, cex = 1)

# 2023
plotRGB(rasters_true_color[[5]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("05.10-06.09", side = 3, line = 1, cex = 1)
mtext("2023", side = 2, line = 2, cex = 1.3, las = 3)
plotRGB(true_color_23625_23725, r = 1, g = 2, b = 3, scale = 10000, stretch = "hist", maxcell = Inf)
mtext("06.25-07.25", side = 3, line = 1, cex = 1)
plotRGB(rasters_true_color[[7]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("08.09-09.09", side = 3, line = 1, cex = 1)
plotRGB(rasters_true_color[[8]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("09.25-10.25", side = 3, line = 1, cex = 1)

# General title
mtext("True Color Sentinel-2", outer = TRUE, side = 3, line = 1, cex = 1.5)

dev.off()
```



- The corrupted images were then composed trough the stack of the single bands

Stack of the band for True colour

- The raster for the band for the Blue, Green and red bands (**b2, b3 and b4**) were loaded and then extracted the single RGB image

- The bands were then stacked and plotted with the **plotRGB()** function

- To ensure a good visualization it was also used the argument **“maxcell = Inf”** that forced the use of all the pixels from the file

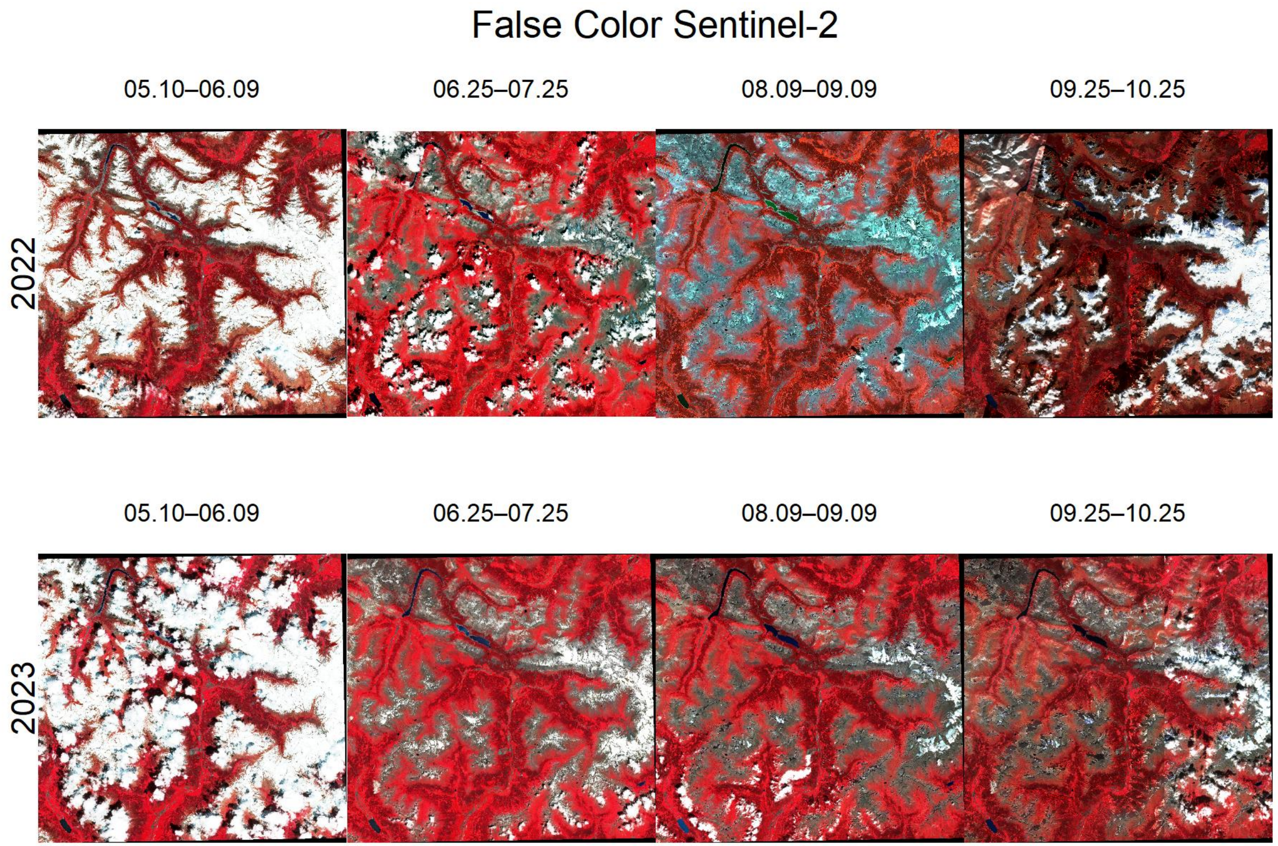
```r
# 3---------------------------------------------------------------------------
rasters_true_color[[3]]
# source: 2022-08-09-00_00_2022-09-09-23_59_Sentinel-2_L2A_True_color.tiff
names(rasters_files[[25]])
names(rasters_files[[26]])
names(rasters_files[[27]])

# Loading rasters corresponding to the bands of true color
b2_2289_2299 <- rasters_files[[25]][[1]]  # Blue
b3_2289_2299 <- rasters_files[[26]][[1]]  # Green
b4_2289_2299 <- rasters_files[[27]][[1]]  # Red

# Extract the first layer of each band to create a single-date RGB image
red_2289_2299 <- b4_2289_2299[[1]]
green_2289_2299 <- b3_2289_2299[[1]]
blue_2289_2299 <- b2_2289_2299[[1]]

# Stack of the RGB bands
true_color_2289_2299 <- c(red_2289_2299, green_2289_2299, blue_2289_2299)

# Plot RGB
plotRGB(true_color_2289_2299, r = 1, g = 2, b = 3, scale=10000, stretch = "hist")

# Plot false color maxcell=inf is needed to force the use of all the pixels from the file
plotRGB(true_color_2289_2299, r = 1, g = 2, b = 3, scale = 10000, stretch = "hist", maxcell = Inf)
```

```r
# 6---------------------------------------------------------------------------
rasters_true_color[[6]]
# source: 2023-06-25-00_00_2023-07-25-23_59_Sentinel-2_L2A_True_color.tiff
names(rasters_files[[61]])
names(rasters_files[[62]])
names(rasters_files[[63]])

# Loading rasters corresponding to the bands of true color
b2_23625_23725 <- rasters_files[[61]][[1]]  # Blue
b3_23625_23725 <- rasters_files[[62]][[1]]  # Green
b4_23625_23725 <- rasters_files[[63]][[1]]  # Red

# Extract the first layer of each band to create a single-date RGB image
red_23625_23725 <- b4_23625_23725[[1]]
green_23625_23725 <- b3_23625_23725[[1]]
blue_23625_23725 <- b2_23625_23725[[1]]

# Stack
true_color_23625_23725 <- c(red_23625_23725, green_23625_23725, blue_23625_23725)

# Plot RGB
plotRGB(true_color_23625_23725, r = 1, g = 2, b = 3, scale=10000, stretch = "hist")

dev.off()
```

6. Visualization of the study area: FALSE COLOUR

- For the false colour it was possible to download 7 images out of the 8 periods, one images were repeatedly **corrupted**:
    - 2022-08-09

```r
# False color final

par(mfrow = c(2, 4), mar = c(1, 2, 3, 1), oma = c(0, 3, 3, 2))

# 2022
plotRGB(rasters_false_color[[1]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("05.10-06.09", side = 3, line = 1, cex = 1)
mtext("2022", side = 2, line = 2, cex = 1.3, las = 3)
plotRGB(rasters_false_color[[2]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("06.25-07.25", side = 3, line = 1, cex = 1)
plotRGB(false_color_2289_2299, r = 1, g = 2, b = 3, scale = 10000, stretch = "lin", maxcell = Inf)
mtext("08.09-09.09", side = 3, line = 1, cex = 1)
plotRGB(rasters_false_color[[4]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("09.25-10.25", side = 3, line = 1, cex = 1)

# 2023
plotRGB(rasters_false_color[[5]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("05.10-06.09", side = 3, line = 1, cex = 1)
mtext("2023", side = 2, line = 2, cex = 1.3, las = 3)
plotRGB(rasters_false_color[[6]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("06.25-07.25", side = 3, line = 1, cex = 1)
plotRGB(rasters_false_color[[7]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("08.09-09.09", side = 3, line = 1, cex = 1)
plotRGB(rasters_false_color[[8]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("09.25-10.25", side = 3, line = 1, cex = 1)

# General title
mtext("False Color Sentinel-2", outer = TRUE, side = 3, line = 1, cex = 1.5)

dev.off()
```



Stack of the band for False colour

- The raster for the band for the Red, Green and NIR (**b4, b3 and b8**) were loaded and then extracted the single RGB image

- The stack worked but the resulting image gives off a blue tint instead of the expected white

- Caused by **reflectance values** are handled in R compared original Sentinel-2

- The blue hue could be resolved with further normalization

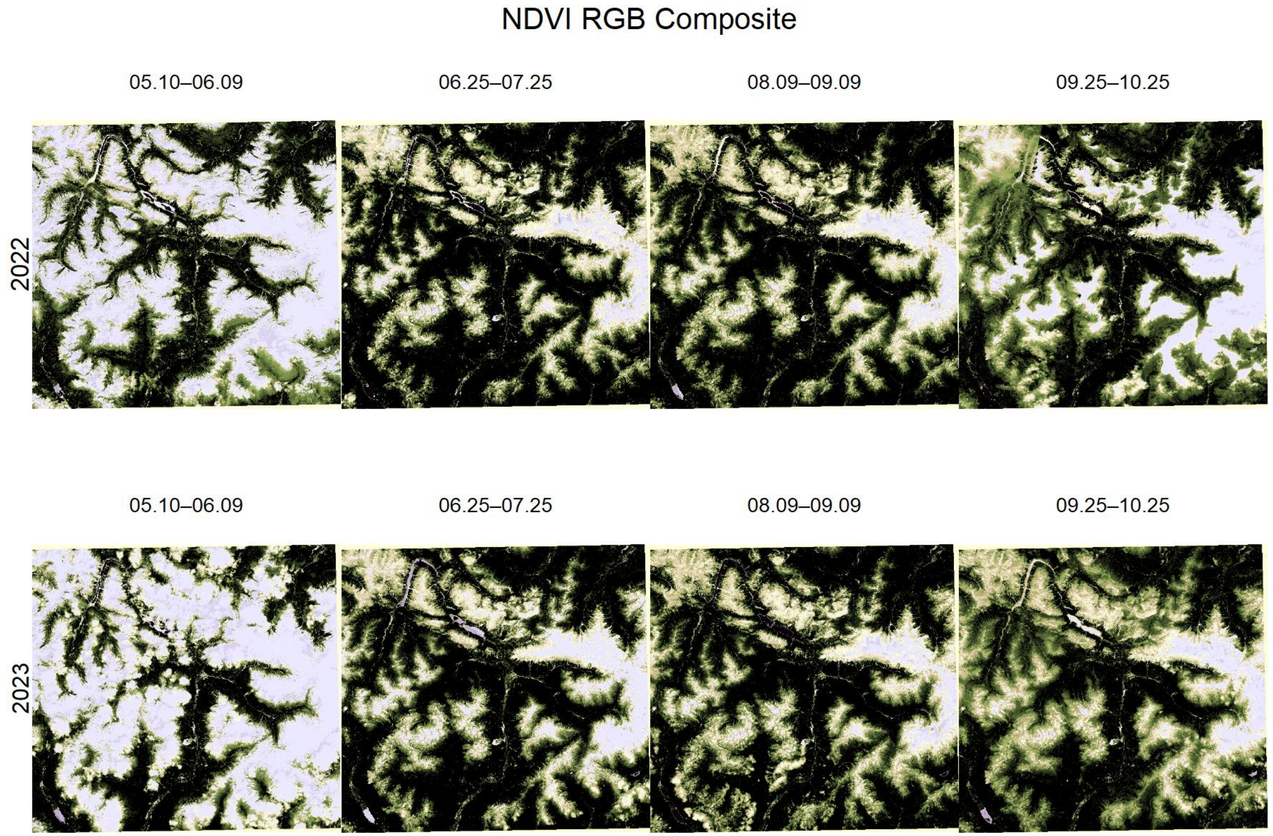
```
# 3---------------------------------------------------------------------------
rasters_false_color[[3]] # Gives us information on the raster
#source: 2022-08-09-00_00_2022-09-09-23_59_Sentinel-2_L2A_False_color.tiff

names(rasters_files[[26]])
names(rasters_files[[27]])
names(rasters_files[[30]])

b4_2289_2299 <- rasters_files[[26]][[1]] # red
b3_2289_2299 <- rasters_files[[27]][[1]] # green
b8_2289_2299 <- rasters_files[[30]][[1]] # (NIR) (8)

# Extract the first layer of each band to create a single-date RGB image
nir_2289_2299 <- b8_2289_2299[[1]]
red_2289_2299 <- b4_2289_2299[[1]]
green_2289_2299 <- b3_2289_2299[[1]]

# Stack of the bands
false_color_2289_2299 <- c(nir_2289_2299, red_2289_2299, green_2289_2299)

# Plot false color maxcell=inf is needed to force the use of all the pixels from the file
plotRGB(false_color_2289_2299, r = 1, g = 2, b = 3, scale = 10000, stretch = "lin", maxcell = Inf)

####
# The image appears with an unusual blue tint instead of the expected white.
# This is likely due to differences in how reflectance values are handled in R
# compared to how they are visualized in the original Sentinel-2 platform.
####
```

7. Visualization of the study area: NDVI COMPOSITE

- For the NDVI images there was no problem with the files

- From the comparison can be seen the rough differences in NDVI

```r
# NDVI already composite final

par(mfrow = c(2, 4), mar = c(1, 2, 3, 1), oma = c(0, 3, 3, 2))

# 2022
plotRGB(rasters_NDVI[[1]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("05.10-06.09", side = 3, line = 1, cex = 0.8)
mtext("2022", side = 2, line = 2, cex = 1, las = 3)
plotRGB(rasters_NDVI[[2]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("06.25-07.25", side = 3, line = 1, cex = 0.8)
plotRGB(rasters_NDVI[[3]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("08.09-09.09", side = 3, line = 1, cex = 0.8)
plotRGB(rasters_NDVI[[4]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("09.25-10.25", side = 3, line = 1, cex = 0.8)

# 2023
plotRGB(rasters_NDVI[[5]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("05.10-06.09", side = 3, line = 1, cex = 0.8)
mtext("2023", side = 2, line = 2, cex = 0.9, las = 3)
plotRGB(rasters_NDVI[[6]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("06.25-07.25", side = 3, line = 1, cex = 0.8)
plotRGB(rasters_NDVI[[7]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("08.09-09.09", side = 3, line = 1, cex = 0.8)
plotRGB(rasters_NDVI[[8]], r = 1, g = 2, b = 3, stretch = "lin")
mtext("09.25-10.25", side = 3, line = 1, cex = 0.8)

# General title
mtext("NDVI RGB Composite", outer = TRUE, side = 3, line = 1, cex = 1.2)

dev.off()
```



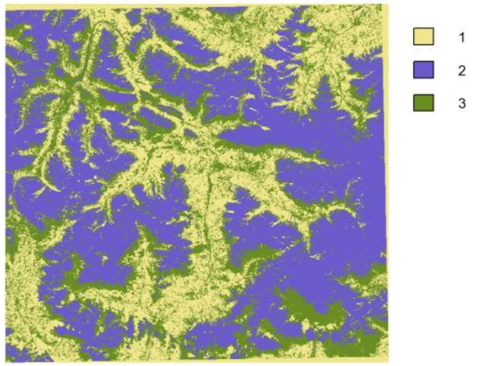
8. Preliminary assessment condition of the forests

- Trough the classification with the **“im.classify()”** function the areas with denser vegetation were highlighted in the true colour images.

- The percentage were then used to **create a data frame** that was then converted in the long format to make it work on with the ggplot function.

- The conversion was carried trough the function “**pivot_longer**” from the “**tidyr**” package.

```r
par(mfrow = c(1,2))
plotRGB(rasters_true_color[[1]], r = 1, g = 2, b = 3, stretch = "lin")
Bands_TC_051022_060922_class <- im.classify(rasters_true_color[[1]], num_clusters = 3)

Freq_TC_051022_060922_class <- freq(Bands_TC_051022_060922_class)
Freq_TC_051022_060922_class
Tot_TC_051022_060922_class <- ncell(Bands_TC_051022_060922_class)
P_TC_051022_060922_class = Freq_TC_051022_060922_class[3]*100/Tot_TC_051022_060922_class #the third column is the count for each pixel
P_TC_051022_060922_class
#1 41.68983 These are mountains' peak
#2 28.56901 These are denser forests
#3 29.74116 These are high-altitude pastures and villages that still have a percentage of cement
```



View of denser vegetation percentage

- The histogram given from the **analysis classification** highlight in a similar way the succession of the denser vegetation.

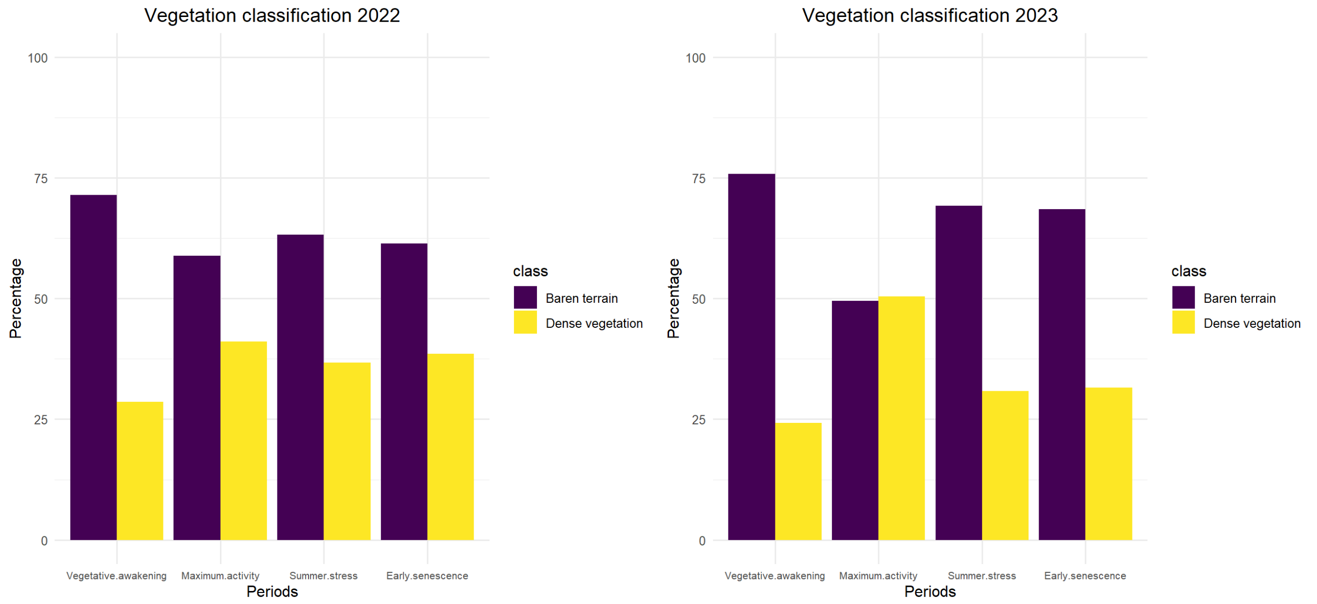
- The **maximum activity** period heighten the amount of denser vegetation while the stress of the month august decreases it

- As a **preliminary analysis** it can give a rough idea on the amount of denser vegetation present but nonetheless useful to direct further analysis.

```r
g2 <- ggplot(df_long23, aes(x = periods, y = percentage, fill = class)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_viridis_d(option = "D") +
  scale_x_discrete(limits = c("Vegetative.awakening",
                              "Maximum.activity",
                              "Summer.stress",
                              "Early.senescence")) + ylim(c(0, 100)) +
  labs(title = "Vegetation classification 2023", x = "Periods", y = "Percentage") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 0, hjust = 0.5, size = 7),
        plot.title = element_text(hjust = 0.5))
```

```r
g1 <- ggplot(df_long22, aes(x = periods, y = percentage, fill = class)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_viridis_d(option = "D") +
  scale_x_discrete(limits = c("Vegetative.awakening",
                              "Maximum.activity",
                              "Summer.stress",
                              "Early.senescence")) + ylim(c(0, 100)) +
  labs(title = "Vegetation classification 2022", x = "Periods", y = "Percentage") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 0, hjust = 0.5, size = 7),
        plot.title = element_text(hjust = 0.5))
```

```r
# Unification of the two plots
g1 + g2
```



9. NDVI and NDWI - Analysis Setup

This section computes two key indices (NDVI and NDWI) using Sentinel-2 imagery collected over four plant development stage between 2022 and 2023.

Bands combined using standard formulas:

NDVI = (NIR - Red) / (NIR + Red)

NDWI = (NIR - SWIR) / (NIR + SWIR)

The spectral bands used are:

B04 (Red) and B08 (NIR) for NDVI → indicates vegetation health

B08 (NIR) and B11 (SWIR) for NDWI → highlights water stress and dryness

```
# From previous errors, some files needed to be reprojected on the CRS
# Mainly the NIR in degrees reprogected on the RED band in meters
nir_projected <- terra::project(nirband08_062522_072522, redband04_062522_072522)
```

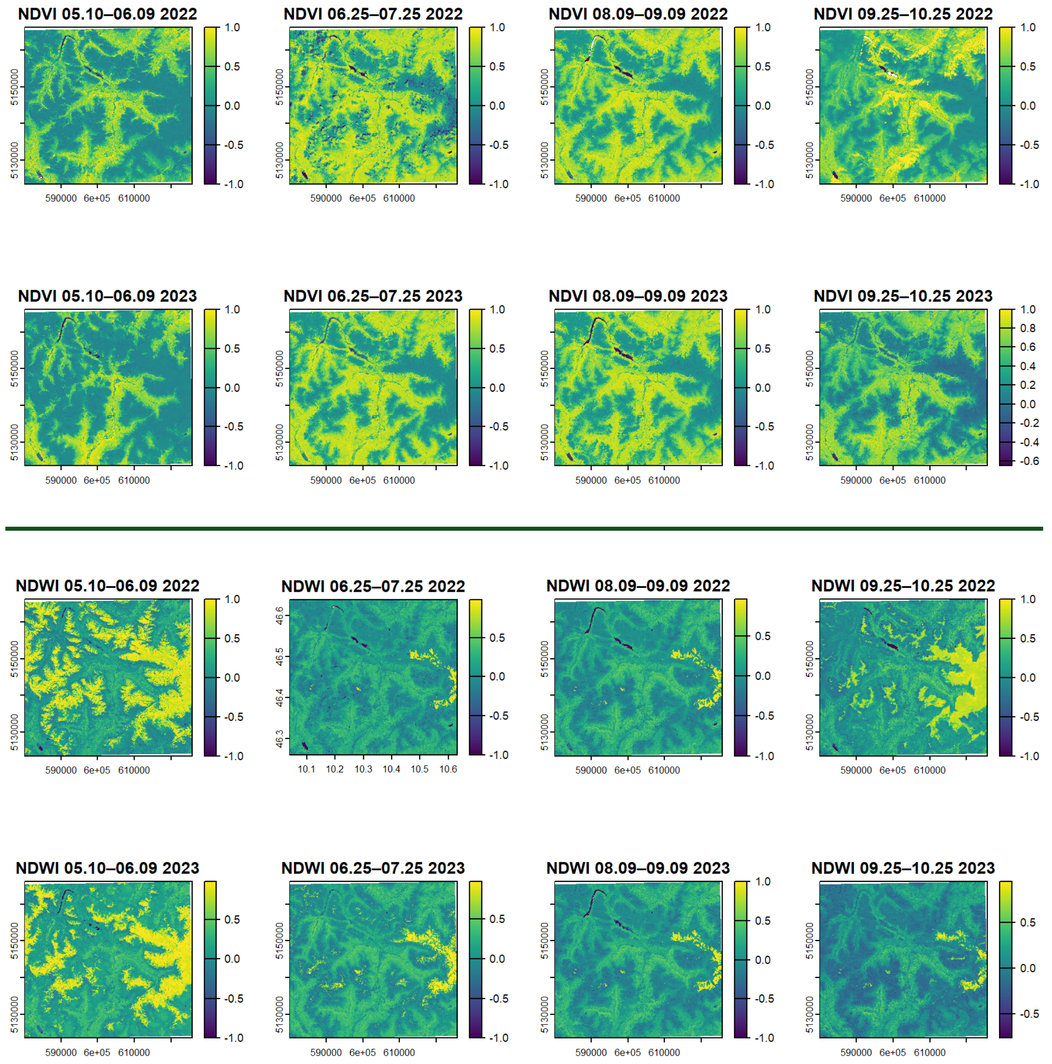
```
# NDVI 2022
NDVI_051022_060922 <- (nirband08_051022_060922 - redband04_051022_060922) / (nirband08_051022_060922 + redband04_051022_060922)
NDVI_062522_072522 <- (nir_projected - redband04_062522_072522) / (nir_projected + redband04_062522_072522)
NDVI_080922_090922 <- (nirband08_080922_090922 - redband04_080922_090922) / (nirband08_080922_090922 + redband04_080922_090922)
NDVI_092522_102522 <- (nirband08_092522_102522 - redband04_092522_102522) / (nirband08_092522_102522 + redband04_092522_102522)

# NDVI 2023
NDVI_051023_060923 <- (nirband08_051023_060923 - redband04_051023_060923) / (nirband08_051023_060923 + redband04_051023_060923)
NDVI_062523_072523 <- (nirband08_062523_072523 - redband04_062523_072523) / (nirband08_062523_072523 + redband04_062523_072523)
NDVI_080923_090923 <- (nirband08_080923_090923 - redband04_080923_090923) / (nirband08_080923_090923 + redband04_080923_090923)
NDVI_092523_102523 <- (nirband08_092523_102523 - redband04_092523_102523) / (nirband08_092523_102523 + redband04_092523_102523)
```

```
# NDWI 2022
NDWI_051022_060922 <- (nirband08_051022_060922 - swirband11_051022_060922) / (nirband08_051022_060922 + swirband11_051022_060922)
NDWI_062522_072522 <- (nirband08_062522_072522 - swirband11_062522_072522) / (nirband08_062522_072522 + swirband11_062522_072522)
NDWI_080922_090922 <- (nirband08_080922_090922 - swirband11_080922_090922) / (nirband08_080922_090922 + swirband11_080922_090922)
NDWI_092522_102522 <- (nirband08_092522_102522 - swirband11_092522_102522) / (nirband08_092522_102522 + swirband11_092522_102522)

# NDWI 2023
NDWI_051023_060923 <- (nirband08_051023_060923 - swirband11_051023_060923) / (nirband08_051023_060923 + swirband11_051023_060923)
NDWI_062523_072523 <- (nirband08_062523_072523 - swirband11_062523_072523) / (nirband08_062523_072523 + swirband11_062523_072523)
NDWI_080923_090923 <- (nirband08_080923_090923 - swirband11_080923_090923) / (nirband08_080923_090923 + swirband11_080923_090923)
NDWI_092523_102523 <- (nirband08_092523_102523 - swirband11_092523_102523) / (nirband08_092523_102523 + swirband11_092523_102523)
```

NDVI and NDWI

- Low values (<0.1) **indicate barren areas** (rock, sand, snow)
- Moderate values (0.2–0.5) reflect **sparse or senescing vegetation.**
- High values (0.6–0.9) correspond to **dense, healthy vegetation.**

- Results are plotted in a comparative layout, allowing for a **visual assessment of seasonal and interannual changes in vegetation activity and water stress.**
- For each period and year, the relevant bands are loaded, reprojected if needed (to align CRS), and combined.



10. Plotting of ΔNDVI and ΔNDWI

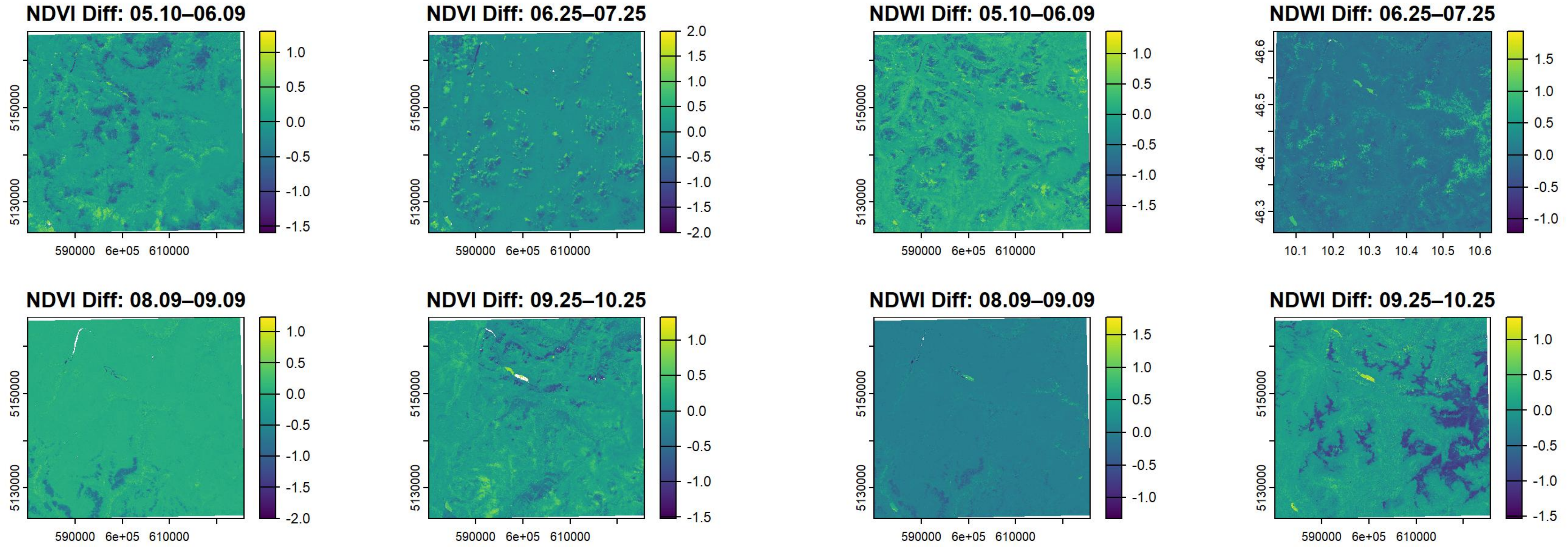
- The result of the plotting were analysed by calculating the ΔNDVI and ΔNDWI. This enabled to highlight the number of **values that changed over the year**.

- NDVI and NDWI values range from -1.0 to +1.0. By calculating the difference (ΔNDVI) between the same phenological periods across two years, we can assess changes in vegetation activity over time (± 2.0).

```
# === NDVI ===
difNDVI_05 <- NDVI_051023_060923 - NDVI_051022_060922
difNDVI_06 <- NDVI_062523_072523 - NDVI_062522_072522
difNDVI_08 <- NDVI_080923_090923 - NDVI_080922_090922
difNDVI_09 <- NDVI_092523_102523 - NDVI_092522_102522
```

```
# === NDWI ===
difNDWI_05 <- NDWI_051023_060923 - NDWI_051022_060922
difNDWI_06 <- NDWI_062523_072523_resampled - NDWI_062522_072522
difNDWI_08 <- NDWI_080923_090923 - NDWI_080922_090922
difNDWI_09 <- NDWI_092523_102523 - NDWI_092522_102522
```

10. Plotting of ΔNDVI and ΔNDWI



ΔNDVI and ΔNDWI distribution

- NDWI and NDVI difference values for each period were extracted and cleaned of missing data.

- These values were then categorized into intervals using a custom classification function.

- **Frequency tables** were created to count how many pixels fall into each category, allowing for a **quantitative assessment** of variation across time.

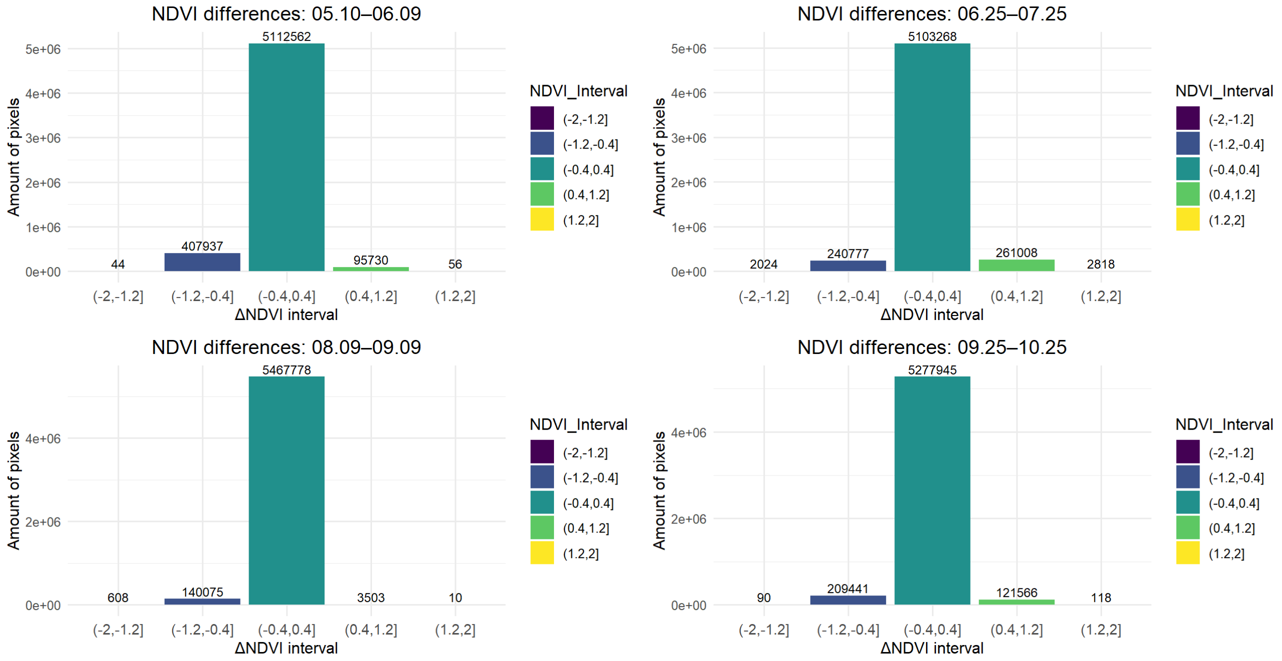
- Factor levels were ordered to ensure correct interval display in plots.

```r
val_difNDVI_05 <- values(difNDVI_05) |>
  na.omit()
val_difNDVI_06 <- values(difNDVI_06) |>
  na.omit()
val_difNDVI_08 <- values(difNDVI_08) |>
  na.omit()
val_difNDVI_09 <- values(difNDVI_09) |>
  na.omit()

summary(val_difNDVI_05)
summary(val_difNDVI_06)
summary(val_difNDVI_08)
summary(val_difNDVI_09)

####
# Function to categorize the NDVI differences based on the limits
categorize_ndvi_diff <- function(x) {
  cut(x, breaks = seq(-2, 2, length.out = 6), right = TRUE)
}
####

dh1 <- categorize_ndvi_diff(val_difNDVI_05)
dh2 <- categorize_ndvi_diff(val_difNDVI_06)
dh3 <- categorize_ndvi_diff(val_difNDVI_08)
dh4 <- categorize_ndvi_diff(val_difNDVI_09)
# Construction of the dataframe
df_dh1 <- as.data.frame(table(dh1))
colnames(df_dh1) <- c("NDVI_Interval", "Count")

df_dh2 <- as.data.frame(table(dh2))
colnames(df_dh2) <- c("NDVI_Interval", "Count")

df_dh3 <- as.data.frame(table(dh3))
colnames(df_dh3) <- c("NDVI_Interval", "Count")

df_dh4 <- as.data.frame(table(dh4))
colnames(df_dh4) <- c("NDVI_Interval", "Count")

# Order for vegetative interval
df_dh1$NDVI_Interval <- factor(df_dh1$NDVI_Interval, levels = levels(dh1), ordered = TRUE)
df_dh2$NDVI_Interval <- factor(df_dh2$NDVI_Interval, levels = levels(dh2), ordered = TRUE)
df_dh3$NDVI_Interval <- factor(df_dh3$NDVI_Interval, levels = levels(dh3), ordered = TRUE)
df_dh4$NDVI_Interval <- factor(df_dh4$NDVI_Interval, levels = levels(dh4), ordered = TRUE)
```

Values distribution of ΔNDVI

ΔNDVI Value	Interpretation
≈ 0	Stable
< 0	Decrease in activity
> 0	Increase in activity
Near ±1 or ±2	Drastic change

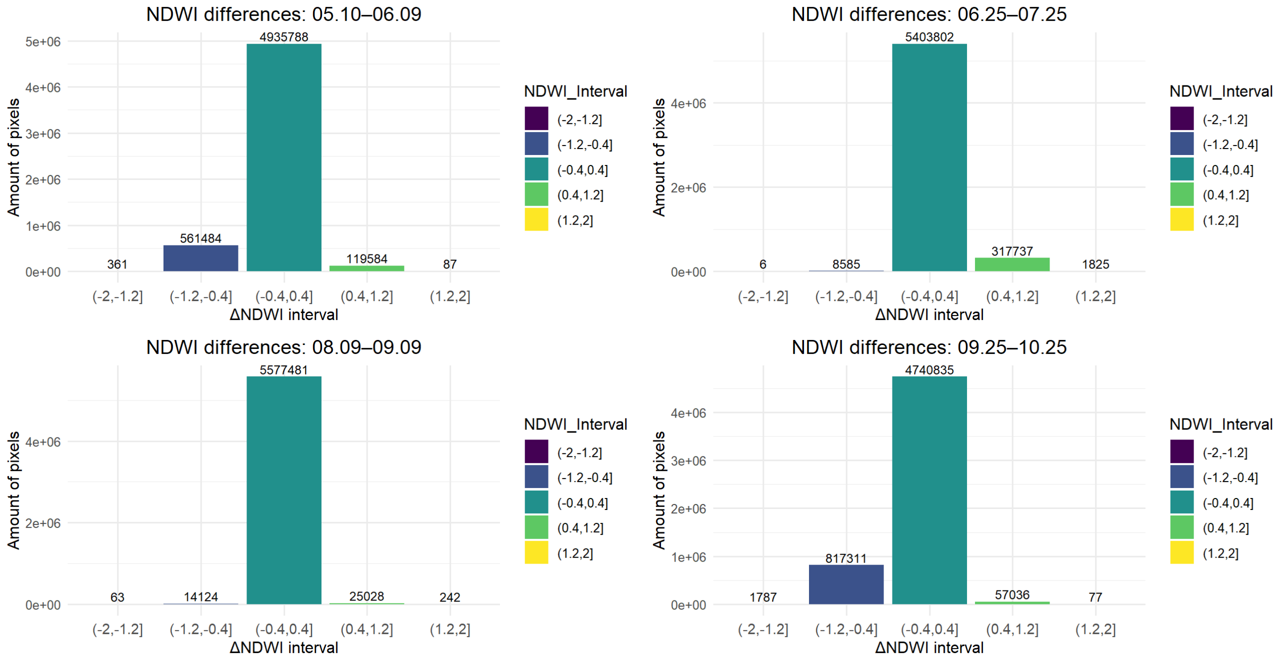
- Most notably the ΔNDVI for the **maximum activity (25.06-25.07)** could support the claim seen from increase in the denser vegetation in 2023 seen from the classification.
- While the interval for the vegetative awakening highlights the **decrease of percentage in 2023** for the denser vegetation.
- The other periods seem undisturbed and in an equilibrium of loss and gain.



ΔNDVI Value	Interpretation
≈ 0	Stable
< 0	Decrease in activity
> 0	Increase in activity
Near ±1 or ±2	Drastic change

Values distribution of ΔNDWI

- Most pixel changes fall within the (-0.4, 0.4] range, likewise for the ΔNDVI, indicating mild seasonal variation between the two years.
- ΔNDWI shows a **bigger loss in the moisture content** of the area for the vegetative awakening and for the early senescence for the year 2023.
- The loss in moisture index in the first and last period of 2023 could be a result of the **delayed drought stress** by the drought in 2022.



11. Statistical analysis through Wilcoxon Rank-Sum Test

- The **Wilcoxon rank-sum test** was used to compare NDVI and NDWI values between corresponding seasonal periods in 2022 and 2023.
- This **non-parametric test** is ideal for detecting shifts in distributions **without assuming normality**, making it suitable for these data.

- Data were extracted from the raster and compared between the same vegetative period.
- The values where then used as input for the Wilcoxon Rank-Sum Test

```
val_NDWI_051022_060922 <- values(NDWI_051022_060922) |>
  na.omit()
```

```
val_NDWI_051023_060923 <- values(NDWI_051023_060923) |>
  na.omit()
```

```
wilcox.test(val_NDWI_051022_060922, val_NDWI_051023_060923, paired = FALSE)
#Wilcoxon rank sum test with continuity correction
#data:  val_NDWI_051022_060922 and val_NDWI_051023_060923
#W = 1.5385e+13, p-value < 2.2e-16
#alternative hypothesis: true location shift is not equal to 0
```

11. Statistical analysis through Wilcoxon Rank-Sum Test

- **NDVI**: Significant differences were found in all seasonal periods ($p < 2.2e-16$), indicating a clear shift in vegetation activity between the two years.
- **NDWI**: All periods show significant differences (three with $p < 2.2e-16$ and one, August–September, with $p = 1.18e-05$), indicating consistent changes in vegetation water content.

- Data were extracted from the raster and compared between the same vegetative period.
- The values where then used as input for the Wilcoxon Rank-Sum Test

```
val_NDWI_051022_060922 <- values(NDWI_051022_060922) |>
  na.omit()
val_NDWI_051023_060923 <- values(NDWI_051023_060923) |>
  na.omit()
wilcox.test(val_NDWI_051022_060922, val_NDWI_051023_060923, paired = FALSE)
#Wilcoxon rank sum test with continuity correction
#data:  val_NDWI_051022_060922 and val_NDWI_051023_060923
#W = 1.5385e+13, p-value < 2.2e-16
#alternative hypothesis: true location shift is not equal to 0
```

12. Conclusions and prospects

- Variations in NDVI, NDWI, and the percentage of dense vegetation highlight significant changes in vegetation cover between 2022 and 2023.
- NDVI values indicate that vegetation was less dense during the early stages of growth in 2023 compared to 2022, possibly due to a delayed drought effect
- Vegetation density increased during the peak of the 2023 growing season, likely as a result of higher average precipitation compared to the previous year.

12. Conclusions and prospects

- The analysis could be improved by applying more advanced statistical methods, such as Principal Component Analysis (PCA), and incorporating detailed meteorological data.

- Extending the time span to five years would enable a more comprehensive assessment of ecosystem conditions and long-term trends.

THANK YOU FOR YOUR ATTENTION

IMPACT OF CLIMATE CHANGE ON MOUNTAINOUS ECOSYSTEMS

Evaluation of the Ecological Effects of Climate Change in the Alta Valtellina Area