# Step 1.4 Analyzing habitat class distribution

Diane ESPEL

2025-07-29

## Contents



## 1 Objectives

This script aims to visualize the distribution of the number of learning plots per habitat class for every typology level and each data source.

## 2 Script explanation

### 2.1 Clean environment and graphics

```
rm(list = ls()) # Clear all objects from the R environment to start fresh
graphics.off()   # Close all graphics devices (if any plots are open)
```

### 2.2 Load required packages

:::

```
library(ggplot2) # For data visualization
```

## 2.3 Create functions

::: {align="justify"} We need to define a "transform_chain()"function that normalizes the nomenclature of habitats: here, adding dots to the toponymy (e.g., A11 –> A.1.1).

This function formats a string by inserting dots between each character. It splits the input string into individual characters, joins them using dots, and returns the resulting formatted string. This is useful for visually separating characters in habitat codes.

```
# Function to add points between character chain
transform_chain <- function(chain) {
  divided_chain <- strsplit(chain, "")[[1]]
  transformed_chain <- paste(divided_chain, collapse = ".")
  return(transformed_chain)
}
```

## 2.4 Define global variables

Note: A global variable is a variable defined outside of any function. This means the variable is accessible from any part of the code, including inside functions. A global variable retains its value throughout the execution of the R script unless it is explicitly modified in the code.

It is important to define at a minimum:

- the "District": the archipelago of interest (e.g. "CRO" for Crozet archipelago)
- the "Island": the island within the archipelago of interest (e.g. "POS" for Possession island)
- the "Year1": the acquisition year of multispectral imagery
- the "maxTypoLevel": the maximum level of habitat classification (i.e. the most accurate level of typology)

```
District='CRO' # 3-letter code for archipelago (e.g. Crozet)
Island='POS'   # 3-letter code for island (e.g. Possession)
Year1="2022"   # Acquisition year of multispectral imagery
maxTypoLevel=4 # Maximum level of habitat classification
```

## 2.5 Set working directory

You must define a general root directory ("localHOME") that serves as the base path for your input and output data. This directory should point to the local environment where:

- input learning plots is located under "data/vector/Plots/PrimaryTypo"
- outputs of plots distribution will be saved under "data/vector/Plots/PrimaryTypo"

```
# Local path
#localHOME = paste0("your_local_path/")
localHOME=paste0("/home/genouest/cnrs_umr6553/despel/CARTOVEGE/")

# path where to open samples data
open_plots_path=paste0(localHOME,"data/vector/Plots/PrimaryTypo")

# path where to save your results
save_plots_path=paste0(localHOME,"data/vector/Plots/PrimaryTypo")
```

## 2.6 Load learning data

```
FILE1=paste0(open_plots_path,"/Quadrats_", District, "_", Island,
 ↪  "_ALL_SOURCES_Centroids_EPSG32739.csv")
learning_points <- read.csv(FILE1, sep=";",dec=".",stringsAsFactors=FALSE)
# `stringsAsFactors=F` ensures character strings don't import as factors
```

## 2.7 Analyze habitat class distribution

The objective here is to calculate the frequency of each habitat class for every typology level and each data source.

To achieve this:

- The frequency of each habitat class is computed
- The distribution is visualized using a bar plot

```
# Define the type of samples plot you have
source_list=c("FIELD","HFI","PHOTO-INTERPRETATION","ALL")


# Run the script for each sample type in plot_list and for each level of classification
for (s in source_list){

  print(paste0("Plot source: ", s))
  if(s=="ALL"){
    learning_points_source=learning_points
  }else{
    learning_points_source=subset(learning_points,learning_points$Source==s)
  }

  # Level loop
  for (l in seq (1:maxTypoLevel)){

    print(paste0("Working at classication level ", l))

    # Create the folder for each level of classification
    LevelFolder=paste0(save_plots_path,"/","Hab_L",l)
    dir.create(LevelFolder,showWarnings=F) # ShowWarnings=F to remove warnings message if file
     ↪  already exists


    # Compute frequency of habitat types  for level of interest
    ihab=which(colnames(learning_points_source)==paste0("Hab_L",l))
    frequency_hab <- as.data.frame(table(learning_points_source[,ihab]))
    frequency_hab[,1]=as.character(frequency_hab[,1])
    colnames(frequency_hab)=c(paste0("Hab_L",l), "Plot_number")

    # Define column of habitat from the frequency_hab dataframe
    Type_habitat=frequency_hab[,1]

    # Bar plot (png format)

↪   NOMpng=paste0(LevelFolder,"/","Nb_quadrats_",District,"_",Island,"_",s,"_SOURCES_level_",l,".png")
    png(file = NOMpng, width = 1000, height = 600)
```

```
    p <- ggplot(frequency_hab, aes(x = Type_habitat, y = Plot_number, fill = Type_habitat)) +
      geom_bar(stat = "identity") + # Create a bar plot
      geom_text(aes(label =Plot_number), vjust = -0.5, color = "black", size = 8) + # Add data
       ↪  labels
      labs(title = paste0("Source :",s, " Plot number per habitat class - Level ", l), # Set plot
       ↪  titles and axis labels
           x = "Habitat class",
           y = "Number of plots") +
      theme_classic() + # Apply classic theme
      theme(axis.text.x = element_text(size = 13, angle = 90, hjust = 1),  # Customize axis labels
            axis.text.y = element_text(size = 13),
            axis.title.x = element_text(size = 16),
            axis.title.y = element_text(size = 16),
            plot.title = element_text(size = 22,face = "bold"),  # Customize plot title
            legend.text = element_text(size = 14),  # Customize legend text
            legend.title = element_text(size = 16,face = "bold")) +  # Customize legend title
      scale_y_continuous(expand = c(0, 0))  +  # Adjust the y-axis range
      labs(fill = "Habitat class")  # Rename the legend title
    print(p)
    dev.off()

    # Bar plot (svg format)
    NOMsvg <-
↪   paste0(LevelFolder,"/Nb_quadrats_",District,"_",Island,"_",s,"_SOURCES_level_",l,".svg")
    svg(file = NOMsvg)
    print(p)
    dev.off()

    # Save frequency of habitat types in a dataframe
    frequency_hab[,1]<- sapply(frequency_hab[,1], transform_chain) # Add points in habitat code
    frequency_hab[,1] <- paste(paste0(District,"-"), frequency_hab[,1], sep = "") # Add CRO- before
↪   characters

     ↪  write.table(frequency_hab,file=paste0(LevelFolder,"/","Nb_quadrats_",District,"_",Island,"_",s,"_SOURCES_le
     ↪  = ";", dec = ".", row.names = FALSE)


  } #end of level loop

} # end of source loop
```

This script calculates and visualizes the number of vegetation plots per habitat class and classification level, across different sampling sources, saving both frequency tables and barplots for each combination.