Wetlands
&
Birds

Federico Tossani

Wintering Birds

25 species you can see during winter time

Who are we looking at?

	species_name	italian_name	pop_trend_period	pop_trend	pop_trend_long_period	pop_trend_long	taxonomic_group
1	Aythya ferina	Moriglione	2009-2015	D	1991-2015	D	Ducks, Geese and Swans
2	Aythya fuligula	Moretta	2009-2015	S	1991-2015	D	Ducks, Geese and Swans
3	Aythya marila	Moretta grigia	2009-2015	D	1991-2015	D	Ducks, Geese and Swans
4	Bucephala clangula	Quattrocchi	2009-2015	D	1991-2015	D	Ducks, Geese and Swans
5	Mergus serrator	Smergo minore	2009-2015	S	1991-2015	D	Ducks, Geese and Swans
6	Calidris minuta	Gambecchio comune	2009-2015	D	1991-2015	D	Waders, Gulls and Auks
7	Hydrocoloeus minutus	Gabbianello	2009-2015	I	1991-2015	D	Waders, Gulls and Auks
8	Podiceps cristatus	Svasso maggiore	2009-2015	S	1991-2015	I	Grebes
9	Anser anser	Oca selvatica	2009-2015	I	1991-2015	I	Ducks, Geese and Swans
10	Tadorna tadorna	Volpoca	2009-2015	I	1991-2015	I	Ducks, Geese and Swans
11	Anas platyrhynchos	Germano reale	2009-2015	D	1991-2015	I	Ducks, Geese and Swans
12	Anas acuta	Codone	2009-2015	I	1991-2015	I	Ducks, Geese and Swans
13	Netta rufina	Fistione turco	2009-2015	I	1991-2015	I	Ducks, Geese and Swans
14	Aythya nyroca	Moretta tabaccata	2009-2015	I	1991-2015	I	Ducks, Geese and Swans
15	Melanitta fusca	Orco marino	2009-2015	D	1991-2015	I	Ducks, Geese and Swans
16	Grus grus	Gru	2009-2015	I	1991-2015	I	Cranes, Rails, Gallinules and Coots
17	Pluvialis apricaria	Piviere dorato	2009-2015	S	1991-2015	I	Waders, Gulls and Auks
18	Pluvialis squatarola	Pivieressa	2009-2015	I	1991-2015	I	Waders, Gulls and Auks
19	Calidris alba	Piovanello tridattilo	2009-2015	I	1991-2015	I	Waders, Gulls and Auks
20	Limosa lapponica	Pittima minore	2009-2015	I	1991-2015	I	Waders, Gulls and Auks
21	Anser albifrons albifrons	Oca lombardella	2009-2015	I	1991-2015	I	Ducks, Geese and Swans
22	Podiceps nigricollis	Svasso piccolo	2009-2015	I	1991-2015	S	Grebes
23	Podiceps auritus	Svasso cornuto	2007-2018	Unk	1991-2015	Unk	Grebes
24	Lymnocryptes minimus	Frullino	2007-2018	Unk	1991-2015	Unk	Waders, Gulls and Auks
25	Anser fabalis rossicus	Oca granaiola	2007-2018	Unk	1993-2018	Unk	Ducks, Geese and Swans

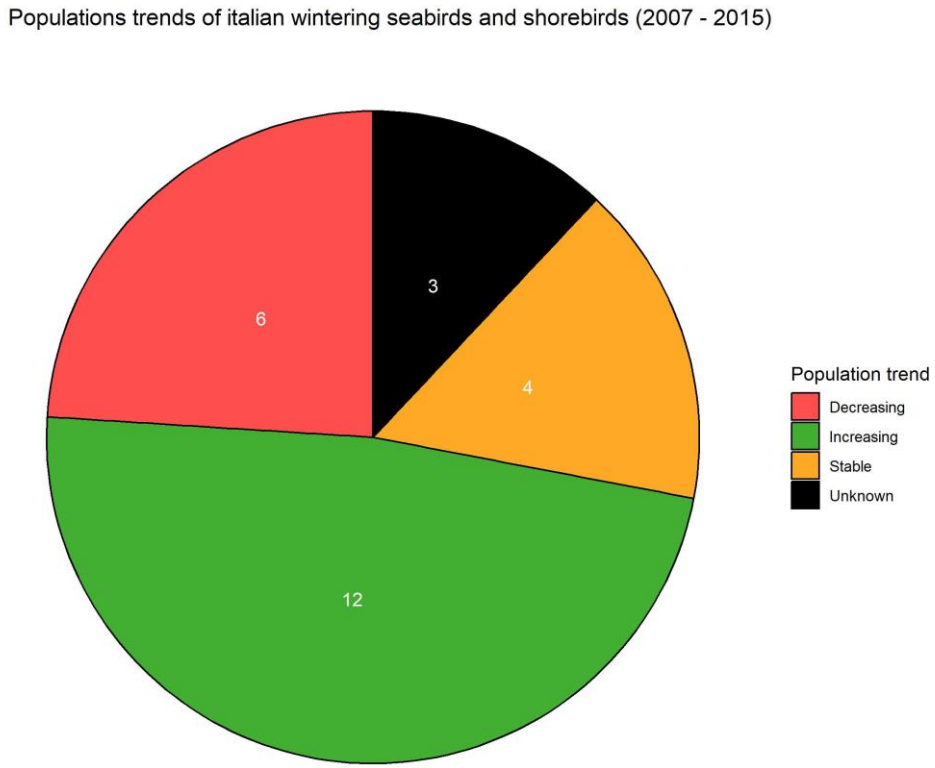
Code

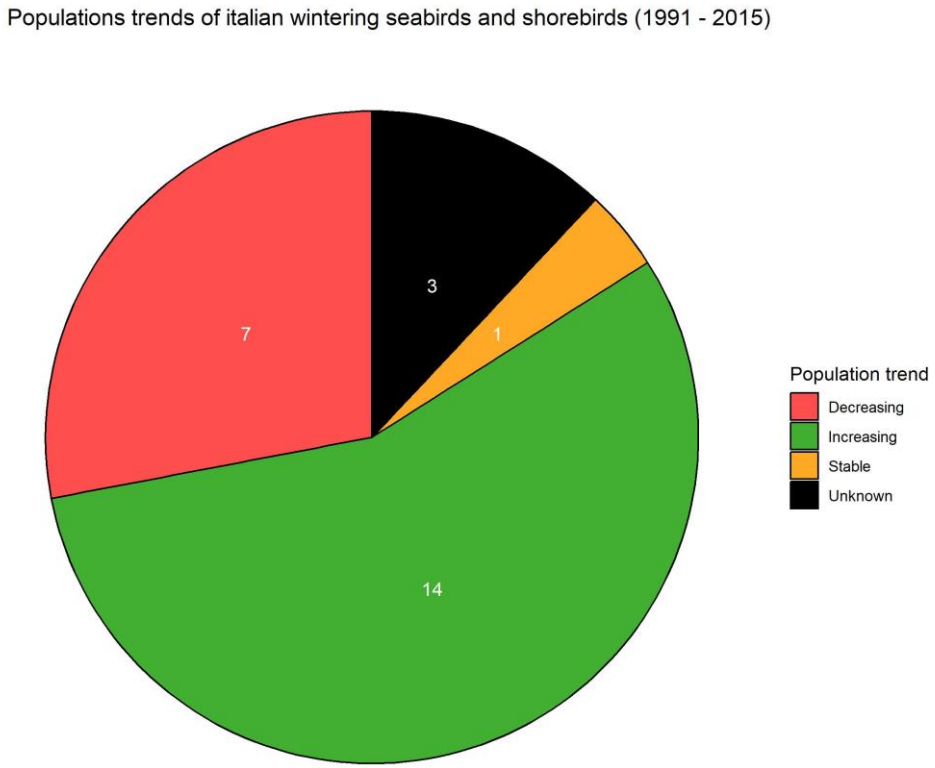
```
source <- data_birds_raw%>%
                dplyr::select(matches("_source"))%>%
                colnames()

data_taxbirds <- data_taxbirds%>%
                     dplyr::select("speciescode", "speciesname","taxOrder", "taxFamily", "taxGroup_en", "taxFamily_en")%>%
                     distinct(.keep_all = TRUE)


variable <- c("country", "season","speciescode", "speciesname", "common_speciesname", "population_date",
              "population_size_unit", "population_size_min", "population_size_max",
              "population_method", "population_trend_period", "population_trend",
              "population_trend_method", "population_trend_long_period", "population_trend_long",
              "population_trend_long_method", "taxGroup_en")

data_birds <- data_birds_raw%>%
                    dplyr::select(-which(names(data_birds_raw) %in% source))%>%
                    filter(country == "IT", season == "W")%>%
                    left_join(data_taxbirds, by = c("speciescode", "speciesname"), keep = F)%>%
                    dplyr::select(all_of(variable))%>%
                    filter(taxGroup_en != "Hawks and Eagles")%>%
                    arrange(speciescode, sort = T)%>%
                    group_by(population_trend)
```

Code

```
data_birds_list <- data_birds%>%
                        dplyr::select("speciesname", "common_speciesname", "population_trend_period", "population_trend",
                                      "population_trend_long_period", "population_trend_long", "taxGroup_en")%>%
                        rename("species_name" = "speciesname",
                               "italian_name" = "common_speciesname",
                               "pop_trend_period" = "population_trend_period",
                               "pop_trend" = "population_trend",
                               "pop_trend_long_period" = "population_trend_long_period",
                               "pop_trend_long" = "population_trend_long",
                               "taxonomic_group" = "taxGroup_en")%>%
                        arrange(pop_trend_long)


stargazer(data_birds_list,
          summary = FALSE,
          type = "text",
          digit.separator = ".",
          out = "wintering_birds.txt")
```

How’s going for birds populations?





Code

```
pop_trend_plot <- ggplot(birds_pop_trend, aes(x = "", y = n, fill = population_trend)) +
                      geom_col(color = "black")+
                      geom_text(aes(label = n),
                                color = "white",
                                position = position_stack(vjust = 0.5)) +
                      coord_polar(theta = "y")+
                      guides(fill = guide_legend(title = "Population trend")) +
                      scale_fill_manual(values = c("#FF4E4E", "#42AF32", "#FFA926", "#000000"),
                                        labels = c("Decreasing", "Increasing", "Stable", "Unknown")) +
                      theme_void()+
                      ggtitle("       Populations trends of italian wintering seabirds and shorebirds (2007 - 2015)")
```

Wetlands

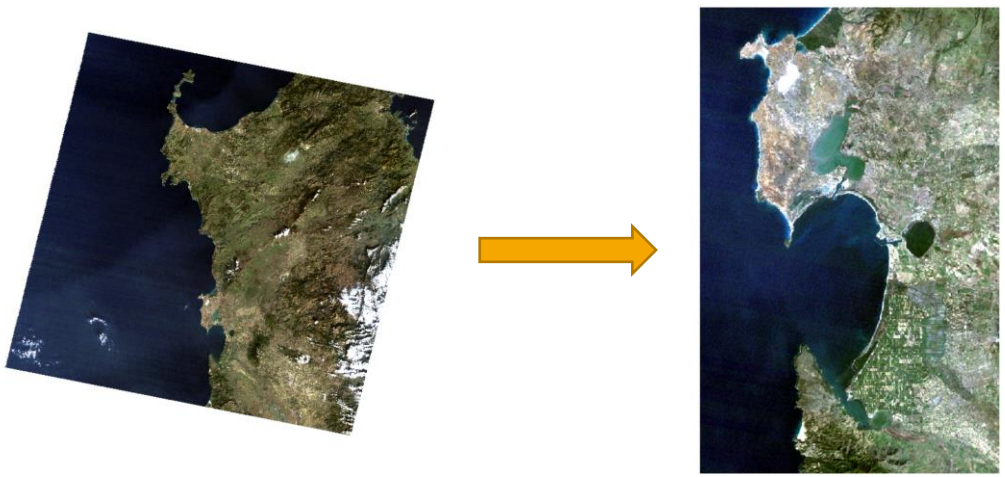
3 coastal wetlands

Load packages

```
# First I create a list of the packages I need

list.of.packages <- c("tidyverse",
                      "raster",
                      "RStoolbox",
                      "viridis",
                      "gridExtra",
                      "stargazer")

# with this line of code I check if alll the packages are installed and then I load it

{
  new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]

  if(length(new.packages)) install.packages(new.packages)

  lapply(list.of.packages, require, character.only = TRUE)
}
```

Image pre-processing

```
list_img <- list.files(pattern = "20200104_B")
img_allbands <- lapply(list_img, raster)
img_stack <- stack(img_allbands)
writeRaster(img_stack, filename="delta_p191r29_20200104.grd", format="raster")
```



```
oristano_list <- list.files(pattern = "oristano_")
oristano_img <- lapply(oristano_list, brick)

# oristano_crop_plot <- plotRGB(oristano_img[[1]], 4, 3, 2, stretch="lin")
# drawExtent(show=TRUE, col="red")

# ext_oristano <- extent(447440, 463826.3, 4411123, 4428401)

crop_file <- function(oristano_img) {

  # Create the extent object
  ext_oristano <- extent(442938.9, 473086.2, 4389311, 4436271)

  # Crop the raster object with the extent
  cropped_img <- crop(oristano_img, ext_oristano)

  # Return the cropped raster object
  return(cropped_img)
}

# Apply the crop function to each file
oristano_cropped <- lapply(oristano_img, crop_file)

# Define the logical index to remove specific elements
remove_index <- c(TRUE, FALSE, TRUE, FALSE, TRUE, FALSE, TRUE, FALSE, TRUE, FALSE)

# Remove the selected elements using the logical index
oristano_cropped <- oristano_cropped[!remove_index]
```

Let’s dive into the False Color World

Oristano 1989
R = NIR, G = Red and B = Green

Cagliari 1989
R = NIR, G = Red and B = Green

Data source: Landsat4 TM

Data source: Landsat4 TM

Oristano 2023
R = NIR, G = Red and B = Green

Cagliari 2022
R = NIR, G = Red and B = Green

Data source: Landsat9 OLI/TIRS

Data source: Landsat9 OLI/TIRS

Po delta 1987
R = NIR, G = Red and B = Green

Data source: Landsat4 TM

Po delta 2020
R = NIR, G = Red and B = Green

Data source: Landsat8 OLI/TIRS

```
# R = nir, G = red, B = green

 delta_plot_nrg1 <- ggRGB(delta_cropped[[1]], 4, 3, 2, stretch = "lin") +
   labs(title = "Po delta 1987",
        subtitle = "R = NIR, G = Red and B = Green",
        caption = "Data source: Landsat4 TM") +
  theme_void()


# R = swir2, G = nir, B = red

delta_plot_s2nr1 <- ggRGB(delta_cropped[[1]], 7, 4, 3, stretch = "lin") +
   labs(title = "Po delta 1987",
        subtitle = "R = SWIR2, G = NIR and B = Red",
        caption = "Data source: Landsat4 TM") +
  theme_void()


nrg_full_plot <- grid.arrange (oristano_plot_nrg1, cagliari_plot_nrg1, delta_plot_nrg1,
                               oristano_plot_nrg5, cagliari_plot_nrg5, delta_plot_nrg5,
                               nrow=2)
ggsave("nrg_plot.jpeg", plot = nrg_full_plot)

# ============================================================ #

s2nr_full_plot <- grid.arrange (oristano_plot_s2nr1, cagliari_plot_s2nr1, delta_plot_s2nr1,
                                oristano_plot_s2nr5, cagliari_plot_s2nr5, delta_plot_s2nr5,
                                nrow=2)
ggsave("s2nr_plot.jpeg", plot = s2nr_full_plot)
```

Oristano 1989
R = SWIR2, G = NIR and B = Red

Cagliari 1989
R = SWIR2, G = NIR and B = Red

Data source: Landsat4 TM

Data source: Landsat4 TM

Oristano 2023
R = SWIR2, G = NIR and B = Red

Cagliari 2022
R = SWIR2, G = NIR and B = Red

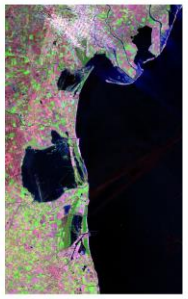
Data source: Landsat9 OLI/TIRS

Data source: Landsat9 OLI/TIRS

Po delta 1987
R = SWIR2, G = NIR and B = Re

Data source: Landsat4 TM

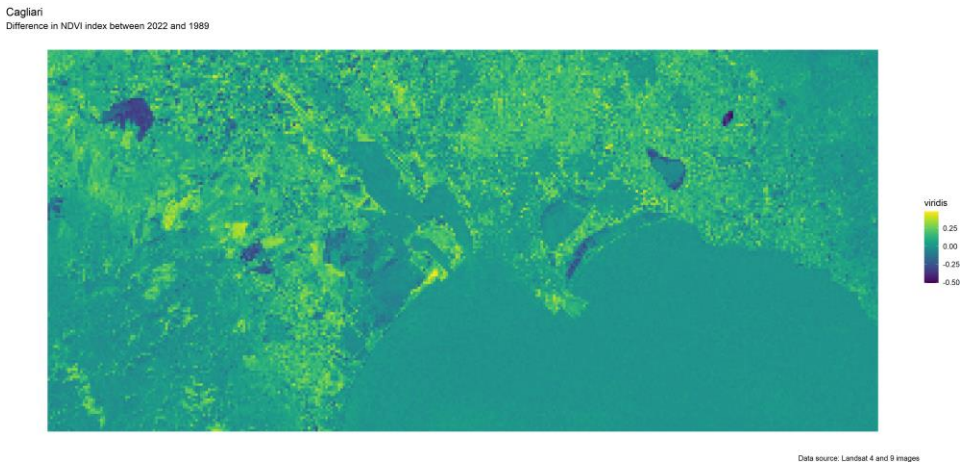
Po delta 2020
R = SWIR2, G = NIR and B = Re



Data source: Landsat8 OLI/TIRS

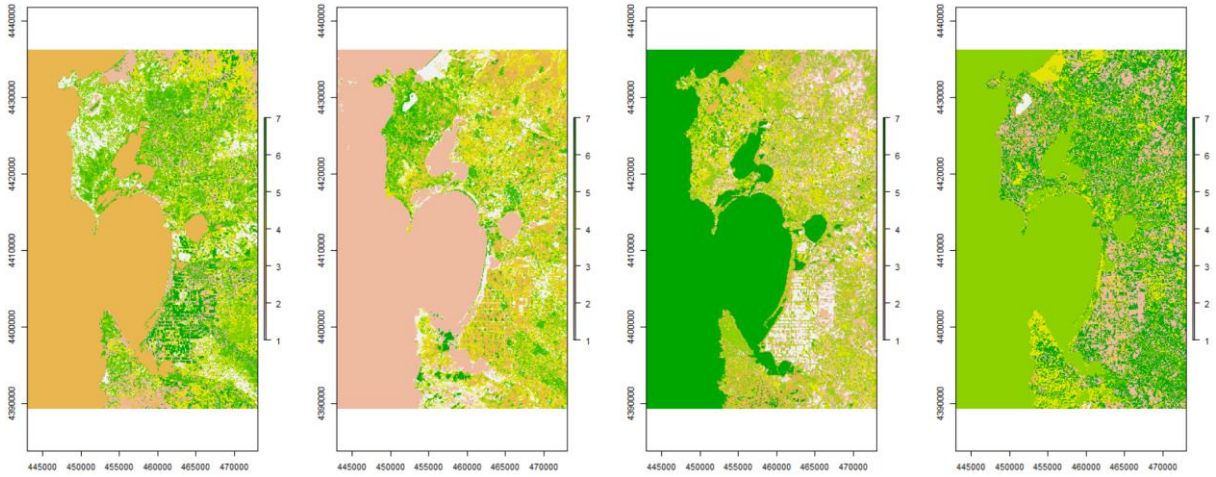
Spectral indicies

```
cagliari89_indices <- RStoolbox::spectralIndices(cagliari_cropped[[1]],
                                                 blue = 1, green = 2, red = 3, nir = 4, swir2 = 5, swir3= 7,
                                                 indices = c("NDVI", "NDWI", "NDWI2", "SLAVI"))

cagliari_ndvi_diff <- cagliari22_indices$NDVI - cagliari89_indices$NDVI
```

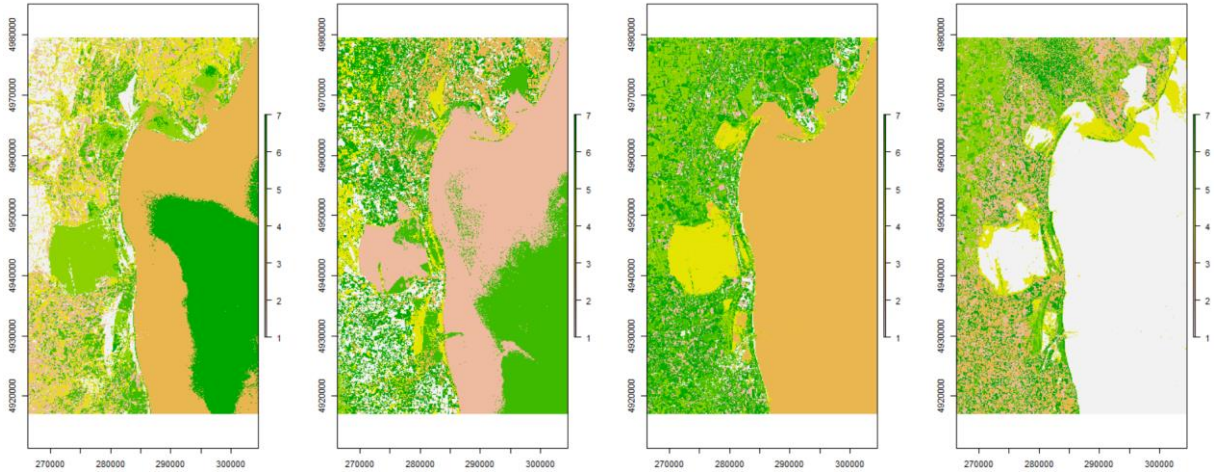


Land Cover Classification

Oristano LCC



Po delta LCC



Land Cover Classification

```
set.seed(999)
oristano89_lcc <- unsuperClass(oristano_cropped[[1]], nSamples = 1000, nClasses = 7)
oristano99_lcc <- unsuperClass(oristano_cropped[[2]], nSamples = 1000, nClasses = 7)
oristano07_lcc <- unsuperClass(oristano_cropped[[3]], nSamples = 1000, nClasses = 7)
oristano17_lcc <- unsuperClass(oristano_cropped[[4]], nSamples = 1000, nClasses = 7)
oristano23_lcc <- unsuperClass(oristano_cropped[[5]], nSamples = 1000, nClasses = 7)

oristano_lcc <- list(oristano89_lcc, oristano99_lcc, oristano07_lcc, oristano17_lcc, oristano23_lcc)

par(mfrow = c(1, 4))
plot(oristano_lcc[[1]]$map)
plot(oristano_lcc[[3]]$map)
plot(oristano_lcc[[4]]$map)
plot(oristano_lcc[[5]]$map)


freq(oristano_lcc[[1]]$map)

#      value  count
# [1,]     1  53309 Agriculture field
# [2,]     2 257931
# [3,]     3 659368 Water
# [4,]     4  30029 Sand and bare soil
# [5,]     5 202108
# [6,]     6 161778
# [7,]     7 208302
```

What’s next?

Improve code aesthetics

(and functionality)

```
# for loop to create all the graph of a region with ggplot2

# Load the data into a list
cagliari_lcc <- list(cagliari84_lcc, cagliari93_lcc, cagliari02_lcc, cagliari11_lcc, cagliari21_lcc)

# Initialize a counter to keep track of the plot number
plot_num <- 1

# Loop through the data in the list
for (i in 1:length(cagliari_lcc)) {
  # Extract the current data
  data <- cagliari_lcc[[i]]

  # Create a plot using ggplot2
  p <- ggplot(data, aes(x = wt, y = mpg)) +
    geom_raster() +
    ggtitle(paste0("Plot ", plot_num))

  # Increment the plot number
  plot_num <- plot_num + 1

  # Show the plot
  print(p)
}
```

```
# for loop to create all the Land Civer Classification of a region with ggplot2

 delta_lcc <- list()

 for (i in 1:length(delta_cropped)) {

   image <- delta_cropped[[i]]

   lcc <- unsuperClass(image, nSamples = 100, nClasses = 5)

   delta_lcc[[i]] <- lcc
 }

# Errore in (function (classes, fdef, mtable)  :
#   non è possibile trovare un metodo ereditato per la funzione ‘nlayers’ per la firma ‘"NULL"’
```

Thank you