Step 3.2 Selecting relevant predictors

3.2.1 Selecting features with PCA

Diane ESPEL

2025-06-18

Contents

1 Objectives

This script aims to perform **predictor selection using Principal Component Analysis (PCA)** on a dataset of raster-derived environmental variables collected from learning plots. The main goal is to identify the most predictive variables by reducing dimensionality and highlighting the variables that contribute most to variance within the dataset. This facilitates downstream modeling by focusing on the most relevant features while minimizing redundancy and multicollinearity among variables.

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(readr)  # For reading delimited text files
library(FactoMineR)  # For running PCA and multivariate analyses
library(factoextra)  # For enhanced PCA visualization (ggplot2-based)
library(corrplot)  # For correlation matrix visualizations
```

2.3 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 `"Satellite1"`: the name of satellite providing multispectral imagery
- the `"Year1"`: the acquisition year of the multispectral imagery

```
District = "CRO"  # 3-letter code for archipelago (e.g. Crozet)
Island = "POS"  # 3-letter code for island (e.g. Possession)
Satellite1 = "Pleiades"  # satellite name of multispectral imagery
Year1 = "2022"  # acquisition year of multispectral imagery
```

2.4 Set working directory

To optimize memory, you must define one general root directory (`"localscratch"`) that serves as the base path for your input and output data, respectively. This directory should point to the local environment where:

- input learning dataset is located under `"data/Learning_data/PrimaryTypo"`
- outputs from PCA will be saved under `"data/Learning_data/PrimaryTypo"`

```
# Base local path (customize to your local environment) localscratch =
# paste0('your_local_path/')
localscratch = paste0("/scratch/despel/CARTOVEGE/")

# Path to open input learning dataset
open_learning_primary_path = paste0(localscratch, "data/Learning_data/PrimaryTypo")

# Path to save results from PCA
save_learning_primary_path = paste0(localscratch, "data/Learning_data/PrimaryTypo")
```

2.5 Load and prepare learning dataset

Before performing PCA, it is often necessary to standardize the data so that each variable has a mean of 0 and a standard deviation of 1. This is especially important when the variables are measured on different scales. To achieve this:

- The script identifies columns corresponding to the first and last raster-derived variables.
- It standardizes the selected variables by centering to zero mean and scaling to unit variance

A **Pearson correlation matrix** is also calculated on the standardized variables to assess relationships and potential multicollinearity. This matrix is saved for further inspection and helps interpret PCA results by revealing clusters of correlated variables.

```
# Open data
FILE1 = paste0(open_learning_primary_path, "/", "Learning_plots_", District, "_",
    Island, "_", Satellite1, "_", Year1, "_ALL_SOURCES_EPSG32739.csv")
learning_data <- read.csv(FILE1, sep = ";", dec = ".", stringsAsFactors = FALSE)  #
 ↪  `stringsAsFactors=F` ensures character strings don't import as factors

# Define Index of variables
imeR = which(colnames(learning_data) == "R")  # First raster-derived variable
imeS = which(colnames(learning_data) == "Slope")  # Last raster-derived variable

# # Standardize variables (mean=0, sd=1)
tab_norm <- as.data.frame(scale(learning_data[, c(imeR:imeS)]))


# Compute correlations between variables
correlation_matrix <- cor(tab_norm)  # Pearson correlations
FILE2 = paste0(save_learning_primary_path, "/", "Correlation_matrix_", District,
    "_", Island, "_", Satellite1, "_", Year1, "_ALL_SOURCES.csv")
write.table(correlation_matrix, FILE2, sep = ";", dec = ",", row.names = FALSE)
```

2.6 Apply PCA

PCA is conducted using the **FactoMineR** package on the standardized variables to reduce dimensionality. The number of principal components is set to the total number of variables.

- The script generates and saves visualizations of **eigenvalue**s (`eig.val`, as variance explained by each component), which help decide how many components to retain.
- **Correlation circles** are plotted to show variable loadings on principal components, illustrating how variables relate to each other and to each component axis. Several versions of these plots are saved, including colored ones highlighting the magnitude of each variable's contribution, aiding in variable interpretation.
- The **quality of representation** (`cos2`) and **contributions** of variables to each principal component are extracted and saved. These metrics quantify how well each variable is represented on the PCA axes and their importance in defining these axes. Contribution plots for the first four components are created to visually identify the most influential variables.
- A threshold for significant contribution is computed, and the **top 10 variables** contributing most to the first two PCA axes are identified and saved. These variables represent key features capturing the main environmental gradients in the dataset.
- PCA scores for individual learning plots are visualized with color gradients reflecting the quality of their representation. These plots help understand the distribution and clustering of samples in the reduced dimensional space.

```r
# Compute PCA  ------------------------------------------------------------

res_pca=PCA(tab_norm,  # dateframe with n rows and p columns
            ncp = ncol(tab_norm), # maximum number of dimensions
            graph = T)# display the graph


# Eigen values  -----------------------------------------------------------

# Get eigen values
eig.val=as.data.frame(res_pca$eig) # Percentage of variance explained
FILE3=paste0(save_learning_primary_path,"/","PCA_",District,"_",Island,"_",Satellite1,"_",Year1,"_ALL_SOURCES_Eigen
write.table(eig.val,FILE3,sep = ";", dec = ",", row.names = FALSE)

# Plot of Eigen values
NOMpng=paste0(save_learning_primary_path,"/","PCA_",District,"_",Island,"_",Satellite1,"_",Year1,"_ALL_SOURCES_Eige
png(file = NOMpng, width = 500, height = 400)
p=fviz_eig(res_pca,title="Eigen values")+
  theme(text = element_text(size = 17),
        axis.title = element_text(size = 17),
        axis.text = element_text(size = 15))
print(p)
dev.off()


# Correlation circles  ----------------------------------------------------------

# Variable graphs
NOMpng=paste0(save_learning_primary_path,"/",
              "PCA_",District,"_",Island,"_",Satellite1,"_",Year1,"_ALL_SOURCES_AllVariables.png")
png(file = NOMpng, width = 700, height = 700)
p=fviz_pca_var(res_pca, repel = T, #repel to avoid label overlaps
               labelsize=3,pointsize=4)+
  ylim (-1,1)+
  xlim(-1,1)+
  theme(text = element_text(size = 17),
        axis.title = element_text(size = 17),
        axis.text = element_text(size = 17))

print(p)
dev.off()

NOMpng=paste0(save_learning_primary_path,"/",
              "PCA_",District,"_",Island,"_",Satellite1,"_",Year1,"_ALL_SOURCES_AllVariables_color.png")
png(file = NOMpng, width = 700, height = 700)
p=fviz_pca_var(res_pca, repel=T, col.var = "contrib",axes=c(1,2),gradient.cols = c("#00AFBB",
 "#E7B800", "#FC4E07"),
               labelsize=3,pointsize=4)+
  ylim (-1,1)+
  xlim(-1,1)+
  theme(text = element_text(size = 17),
        axis.title = element_text(size = 17),
        axis.text = element_text(size = 17))
print(p)
dev.off()
```

```
NOMsvg=paste0(save_learning_primary_path,"/",
              "PCA_",District,"_",Island,"_",Satellite1,"_",Year1,"_ALL_SOURCES_AllVariables_color.svg")
svg(file = NOMsvg)
print(p)
dev.off()


NOMpng=paste0(save_learning_primary_path,"/",
              "PCA_",District,"_",Island,"_",Satellite1,"_",Year1,"_ALL_SOURCES_AllVariables.png")
png(file = NOMpng, width = 700, height = 700)
p=plot(res_pca,cex=1,cex.axis=1.8,font.axis=1.5)
print(p)
dev.off()

NOMsvg=paste0(save_learning_primary_path,"/",
              "PCA_",District,"_",Island,"_",Satellite1,"_",Year1,"_ALL_SOURCES_AllVariables.svg")
              #svg format to modify into Inkscape
svg(file = NOMsvg)
print(p)
dev.off()



# Graphs of all variables  ------------------------------------------------------------

#  get results for variables metrics (cos2, contribution, etc.)
var=get_pca_var(res_pca) # for both quantitative and qualitative variables

# Quality of the representation
Quality=as.data.frame(var$cos2)
FILE4=paste0(save_learning_primary_path,"/",
             "PCA_",District,"_",Island,"_",Satellite1,"_",Year1,"_ALL_SOURCES_AllVariables_quality_of_represen
write.table(Quality,FILE4,sep = ";", dec = ",", row.names = FALSE)

NOMpng=paste0(save_learning_primary_path,"/",
              "PCA_",District,"_",Island,"_",Satellite1,"_",Year1,"_ALL_SOURCES_Corrplot_Quantitative_Variables
png(file = NOMpng, width = 700, height = 700)
corrplot(na.omit(var$cos2), is.corr=FALSE,tl.cex=1.4,tl.col = "black",cl.cex=1.4,cl.align.text="l")
 # show the link between variables , representation quality and correlations with variables and
 dimensions
dev.off()


#All variables Contributions to dimensions
Var_contrib=as.data.frame(var$contrib)
FILE5=paste0(save_learning_primary_path,"/",
             "PCA_",District,"_",Island,"_",Satellite1,"_",Year1,"_ALL_SOURCES_AllVariables_contributions_to_ax
write.table(Var_contrib,FILE5,sep = ";", dec = ",", row.names = FALSE)

for (dim in seq(1:4)){
  NOMpng=paste0(save_learning_primary_path,"/",
                "PCA_",District,"_",Island,"_",Satellite1,"_",Year1,"_ALL_SOURCES_Variables_contributions_DIM",
  png(file = NOMpng, width = 500, height = 400) # doted line show average value if contributions
   were uniform
```

```
  p=fviz_contrib(res_pca, "var", axes = dim)+ # Contribution to 1st dimension (DIM1)
    theme(text = element_text(size = 17),
          axis.title = element_text(size = 17),
          axis.text = element_text(size = 10))
  print(p)
  dev.off()
}

print(paste0("Contributive threshold : ", round(((1/length(tab_norm))*10^2),2)," %"))

Top10=fviz_contrib(res_pca, choice = "var", axes = 1:2, top =10,sort.val="desc")
Top10=as.data.frame(Top10[["layers"]][[1]][["data"]][["name"]]) # extract the names of the most
 ↪  contributives variables
names(Top10)[1] <- "variables" # Rename first column
FILE6=paste0(save_learning_primary_path,"/",

             ↪  "PCA_",District,"_",Island,"_",Satellite1,"_",Year1,"_ALL_SOURCES_10_most_contributives_variables.
write.table(Top10,FILE6,sep = ";", dec = ",", row.names = FALSE)


# Individuals graphs----------------------------------------------------------

# Plot individuals with cos2 coloring
NOMpng=paste0(save_learning_primary_path,"/",
              "PCA_",District,"_",Island,"_",Satellite1,"_",Year1,"_ALL_SOURCES_Individuals.png")
png(file = NOMpng, width = 700, height = 700)
p=fviz_pca_ind(res_pca, repel=TRUE, col.ind = "cos2", gradient.cols = c("#00AFBB", "#E7B800",
 ↪  "#FC4E07"),labelsize=5,pointsize=4,ggrepel = TRUE, geom = c("text","point"))+
  ylim (-7.5, 7.5)  + xlim(-10,10) +
  theme(text = element_text(size = 17),
        axis.title = element_text(size = 17),
        axis.text = element_text(size = 15))
print(p)

dev.off()


NOMpng=paste0(save_learning_primary_path,"/",
              "PCA_",District,"_",Island,"_",Satellite1,"_",Year1,"_ALL_SOURCES_Individuals2.png")
png(file = NOMpng, width = 700, height = 700)
p=plot(res_pca, choix = "ind",ylim=c(-7.5,7.5),xlim=c(-10,10))
print(p)
dev.off()

NOMsvg=paste0(save_learning_primary_path,"/",
              "PCA_",District,"_",Island,"_",Satellite1,"_",Year1,"_ALL_SOURCES_Individuals2.svg")
               ↪  #svg format
svg(file = NOMsvg)
print(p)
dev.off()
```

2.7 Variable selection from dataset

Variables contributing more than **20%** cumulatively to the first two principal components are selected as the most relevant predictors. The dataset is filtered to retain only these variables, and both the full and filtered datasets (excluding plots from photo-interpretation sources) are saved for use in subsequent modeling or analysis.

```r
# Select variables contributing > 20% to first two axes
total_contrib <- rowSums(var$contrib[, 1:2])
selected_vars <- names(total_contrib[total_contrib > 20])  # 20 % threshold
data_selected <- learning_data[, selected_vars]
FILE7 = paste0(save_learning_primary_path, "/Selected_learning_plots_", District,
    "_", Island, "_", Satellite1, "_", Year1, "_ALL_SOURCES_EPSG32739.csv")
write.table(data_selected, file = FILE7, sep = ";", dec = ".", row.names = FALSE)

# save filtered dataset
true_learning_plots = subset(data_selected$Source != "PHOTO-INTERPRETATION")
FILE8 = paste0(save_learning_primary_path, "/Selected_learning_plots_", District,
    "_", Island, "_", Satellite1, "_", Year1, "_TRUE_SOURCES_EPSG32739.csv")
write.table(true_learning_plots, file = FILE8, sep = ";", dec = ".", row.names = FALSE)
```