

Appendix 4 | ES bundles analysis R Code

```
# #####  
# Script to ES bundles in the EDM region  
#Author: A. Buchadas  
# Run around the period of: 2020-09  
# #####  
  
##### LOAD REQUIRED PACKAGES #####  
  
library(tidyr)  
library(ggplot2)  
library(ggpubr)  
library(dplyr)  
library(mclust)  
library(vegan)  
library(cIValid)  
library(devtools)  
library(ggfortify)  
library(factoextra)  
library(NbClust)  
library(usdm)  
library(PerformanceAnalytics)  
  
#### SET INPUT FOLDER ####  
  
setwd("../R scripts ES/")  
  
#### functions needed ####  
  
norm = function(x){ (x-min(x))/(max(x)-min(x))}  
  
#####Variables#####  
  
ES<-read.csv("../R scripts ES/data/responvariselect.csv")  
drivers<- read.csv("../R scripts ES//data/Explanatoryfinal3.csv")  
dt <- inner_join(ES, drivers, by = "DICOFRE")  
rv1=ES  
rv1s<-ES %>% mutate_at(c(2:9), ~(scale(.)))  
rownames(rv1s) <- rv1s[,1]  
rv1s <- rv1s[,-1]
```

```

ev=drivers
evs <- ev %>% mutate_at(c(2:7), ~(scale(.)))
rownames(evs) <- evs[,1]
evs <- evs[,-1]
evs= as.data.frame(evs)

#####VIF#####
vif(rv1s)
vif(evs)

#####Spearman correlation#####
CorAllvaria<-cor(rv1s, method = c("spearman"))
CorAllvaria
chart.Correlation(rv1s, histogram=TRUE, method = c("spearman"))
chart.Correlation(evs, histogram=TRUE, method = c("spearman"))

#####PCA#####
PCArv1= prcomp(rv1s, center= T, scale = T)

png(file = "ESfarmb_foodc/PCAFinal_3092020.png", width = 1800, height = 1800, units = "px",pointsize = 12,
antialias="default", res = 300)

autoplot(PCArv1, data = PCArv1, colour = 'light grey', loadings = TRUE,loadings.colour = 'black',loadings.alpha = 0.2,
         loadings.label = TRUE, loadings.label.size = 6, loadings.label.repel=T, loadings.label.vjust = 1.2, loadings.label.colour=
"#CD3333", scale = 0) +theme_bw() + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
theme(axis.title.y = element_text(size = 12, angle = 90)) + theme(axis.title.x = element_text(size = 12, angle = 00))+
geom_hline(yintercept = 0, linetype = "dashed") + geom_vline(xintercept = 0, linetype = "dashed")

dev.off()

#Scores for clustering analysis
scores<- PCArv1$x
scoresPC<- as.data.frame(scores)
scoresPC=scoresPC[,1:3]
summary(PCArv1)
get_eigenvalue(PCArv1)
write.csv(scoresPC, file = ".../R scripts ES/ESfarmb_foodc/scoresPC19092020.csv")

```

```
#####RDA####
```

```
var_rda <- rda(rv1s ~., data = evs, scale = TRUE)

png(file = "ESfarmb_foodc/RDAFINAL19092020.png", width = 1800, height = 1800, units = "px",pointsize = 12,
antialias="default", res = 300)

plot(var_rda, type="n",xlim=c(-1,2.5),ylim=c(-2,1))
points(var_rda, pch=1, col="grey", cex=0.5)
arrows(0, 0, var_rda$CCA$biplot[,1], var_rda$CCA$biplot[,2], length = 0.08, angle = 30, col = "black")
text(var_rda$CCA$biplot[,1:2], pch=24, labels=rownames(var_rda$CCA$biplot[,1:2]), col="black", cex=0.9, font=1, pos=2)
text(var_rda, pch=24, dis="sp", col="#CD3333", cex=0.9, pos=3)
points(var_rda, pch=2, dis="sp", col="#CD3333", cex=0.5)
dev.off()
summary(var_rda)
```

```
# Extraction of canonical coefficients from rda object
```

```
coef(var_rda)
```

```
# Tests of all canonical axes
```

```
set.seed(111)
anova.cca(var_rda, by = "axis", step = 1000)
```

```
#####Test clustering algorithms and number of clusters - clvalid#####
```

```
#method ward
```

```
intern <- clValid(scoresPC, 2:6, clMethods = c("hierarchical", "kmeans", "diana", "model", "sota", "pam", "clara", "agnes"),
validation = "internal",maxitems = 6000, metric = "euclidean", method = "ward")
summary(intern)
```

```
stab <- clValid(scoresPC, 2:6, clMethods = c("hierarchical", "kmeans", "diana", "model", "sota", "pam", "clara", "agnes"),
validation = "stability",maxitems = 6000, metric = "euclidean", method = "ward")
summary(stab)
```

```
#####selected cluster: KMEANS 4####
```

```
Kmeans4_PCArv1<-kmeans(scoresPC, 4, nstart = 100)
Cluster4= as.data.frame(Kmeans4_PCArv1$cluster)
Cluster4 <- tibble::rownames_to_column(Cluster4, "DICOFRE")
Cluster4$DICOFRE=as.integer(Cluster4$DICOFRE)
names(Cluster4)[2] <- "Kmeans4"
rv1sc <- tibble::rownames_to_column(rv1s, "DICOFRE")
rv1sc$DICOFRE=as.integer(rv1sc$DICOFRE)
ESKM <- inner_join(rv1sc, Cluster4, by = "DICOFRE")
```

#####ESBoxplot4#####Figure 4#####

rgb2hex(136,203,102)#1- "#88cb66"

rgb2hex(240,148,102)#2- "#f09466"

rgb2hex(163,173,225)#3- "#a3ade1"

rgb2hex(237,237,237)#4 -"#ededed"

rgb2hex(191,191,191)#4 -"#bfbfbf"

png(file = "ESfarmb_foodc/boxplots4kmeES_clor.png", width = 1600, height = 1200, units = "px",pointsize = 10, antialias="default", res = 300)

par(mfrow = c(2,4), mai = c(0.2,0.2,0.2,0.2))

boxplot(Carbseq~Kmeans4,

data=ESKM,

main="Carbseq",

xlab="bundle",

notch=TRUE,

outline = FALSE,

col=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),

medcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),

whiskcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),

staplecol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),

boxcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf")

)

boxplot(ErsPr~Kmeans4,

data=ESKM,

main="ErsPr",

xlab="bundle",

notch=TRUE,

outline = FALSE,

col=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),

medcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),

whiskcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),

staplecol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),

boxcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf")

)

```
boxplot(Pollin~Kmeans4,  
  data=ESKM,  
  main="Pollin",  
  xlab="bundle",  
  notch=TRUE,  
  outline = FALSE,  
  col=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),  
  medcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),  
  whiskcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),  
  staplecol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),  
  boxcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf")  
)
```

```
boxplot(NatTour~Kmeans4,  
  data=ESKM,  
  main="NatTour",  
  xlab="bundle",  
  notch=TRUE,  
  outline = FALSE,  
  col=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),  
  medcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),  
  whiskcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),  
  staplecol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),  
  boxcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf")  
)
```

```
boxplot(FarmBirds~Kmeans4,  
  data=ESKM,  
  main="FarmBirds",  
  xlab="bundle",  
  notch=TRUE,  
  outline = FALSE,  
  col=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),  
  medcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),
```

```
whiskcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),
staplecol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),
boxcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf")
```

```
)
```

```
boxplot(Foodcrops~Kmeans4,
```

```
  data=ESKM,
  main="FoodCrops",
  xlab="bundle",
  notch=TRUE,
  outline = FALSE,
  col=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),
  medcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),
  whiskcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),
  staplecol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),
  boxcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf")
```

```
)
```

```
boxplot(Wine~Kmeans4,
```

```
  data=ESKM,
  main="Wine",
  xlab="bundle",
  notch=TRUE,
  outline = FALSE,
  col=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),
  medcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),
  whiskcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),
  staplecol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),
  boxcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf")
```

```
)
```

```
boxplot(Cattle~Kmeans4,
```

```
  data=ESKM,
  main="Cattle",
  xlab="bundle",
```

```

notch=TRUE,
outline = FALSE,
col=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),
medcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),
whiskcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),
staplecol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf"),
boxcol=c("#88cb66", "#f09466", "#a3ade1", "#bfbfbf")
)
par(mfrow = c(1,1))
dev.off()

```

#####multinomial logistic regression_ to relate drivers with clusters####

```

ESKM$Kmeans4<-factor(ESKM$Kmeans4)
ESKM <- inner_join(ESKM, drivers, by = "DICOFRE")

```

#####Kmeans41 relevel

```

ESKM=as.data.frame(ESKM)
ESKM$out<-relevel(ESKM$Kmeans4, ref="1")
mymodelt4_1<-multinom(out~SEI+ED+FarmerN+Farmsize+SpInd+ProdValue, data=ESKM)
summary(mymodelt4_1)

```

#2-tailed z test

```

zt4_1 <- summary(mymodelt4_1)$coefficients/summary(mymodelt4_1)$standard.errors
ptF4_1 <- (1 - pnorm(abs(zt4_1), 0, 1)) * 2
ptF4_1
write.csv(ptF4_1, file = "../R scripts ES/multino/ptF4_1.csv")
write.csv(cmtF5_1, file = "../R scripts ES/multino/cmtF5_1.csv")

```

#####Kmeans42 relevel

```

ESKM$out2<-relevel(ESKM$Kmeans4, ref="2")
mymodelt4_2<-multinom(out2~SEI+ED+FarmerN+Farmsize+SpInd+ProdValue, data=ESKM)
summary(mymodelt4_2)

```

#2-tailed z test

```
zt4_2 <- summary(mymodelt4_2)$coefficients/summary(mymodelt4_2)$standard.errors
ptF4_2 <- (1 - pnorm(abs(zt4_2), 0, 1)) * 2
ptF4_2
write.csv(ptF4_2, file = "../R scripts ES/multino/ptF4_2.csv")
```

```
#####Kmeans44 relevel
```

```
ESKM$out4<-relevel(ESKM$Kmeans4, ref="3")
mymodelt4_3<-multinom(out4~SEI+ED+FarmerN+Farmsize+SpInd+ProdValue, data=ESKM)
summary(mymodelt4_3)
```

```
#2-tailed z test
```

```
zt4_3 <- summary(mymodelt4_3)$coefficients/summary(mymodelt4_3)$standard.errors
ptF4_3 <- (1 - pnorm(abs(zt4_3), 0, 1)) * 2
ptF4_3
write.csv(ptF4_3, file = "../R scripts ES/multino/ptF4_3.csv")
```

```
#####Kmeans44 relevel
```

```
ESKM$out4<-relevel(ESKM$Kmeans4, ref="4")
mymodelt4_4<-multinom(out4~SEI+ED+FarmerN+Farmsize+SpInd+ProdValue, data=ESKM)
summary(mymodelt4_4)
```

```
#2-tailed z test
```

```
zt4_4 <- summary(mymodelt4_4)$coefficients/summary(mymodelt4_4)$standard.errors
ptF4_4 <- (1 - pnorm(abs(zt4_4), 0, 1)) * 2
ptF4_4
write.csv(ptF4_4, file = "../R scripts ES/multino/ptF4_4.csv")
```

```
#####HNVf#####joining HNVf with cluster results#####
```

```
bundleHNVft<-read.csv("../R scripts ES/bundleHNVft_link1.csv")#####this needs the dataset on percentage of HNVf per parish_
treated aside
```

```
bundleHNVft= bundleHNVft[2:6]
```

```
ESKMhmv<- inner_join(ESKM, bundleHNVft, by = "DICOFRE")
```



```
write.csv(ESKM, file = ".../R scripts ES/ESfarmb_foodc/ESKM_3009hmv_results.csv")
```

```
##### Multifuncionalidade per cluster#####
```

```
ESn = ES[2:9] %>% mutate_at(c(1:8), ~(norm(.)))
```

```
ESstest <-ESKM %>% mutate(simpson=diversity(ESn, index = "simpson", MARGIN = 1, base = exp(1)))
```

```
####getting mean and SD of H index####
```

```
ESKM= ESKM %>% mutate(H= 1/(1-simpson))
```

```
###Summarizing multifunctionality
```

```
meanmultifunctionality = ESKM %>%
```

```
  group_by(Kmeans4) %>%
```

```
  summarise_all(funs(mean))
```

```
sdnmultifunctionality = ESKM %>%
```

```
  group_by(Kmeans4) %>%
```

```
  summarise_all(funs(sd))
```