code\_basic

Yasmin Benitez Ortiz

2025-10-09

# **Icterus proyect**

#### Librerías.

Herramienta para gestión de librerías, una de sus funciones es p\_load, que revisa si los paquetes están instalados y de no estarlo los carga automáticamente.

#install.packages("pacman")  
library("pacman")  
p\_load("tidyverse",  
 "readr",  
 "here",  
 "spThin",  
 "readr",  
 "dplyr",  
 "dendextend")

## Generación de dendograma

“read\_csv” (tidy) crea un objeto leyendo la base de datos csv obtenida con la limpieza en openrefine y qgis, %>% (tidy) encadena la operación, pasando el resultado al siguiente proceso con la librería here para ubicar el archivo

se descartan los vectores (columnas) que no aportan información útil para la creación del clúster aplicando “select” (tidy).

“scale” escala\*\*\* para poder realizar el cálculo de las distancias y se coerciona con “as\_tibble” (tidy) para un manejo más limpio de los datos

crea una matriz de distancias por distancias euclidianas con “dist”

“hclust” crea un clúster\_\*\*\*Agrupamiento por ligamiento medio o promedio: calcula todas las disimilitudes por pares entre los elementos del grupo 1 y los elementos del grupo 2, y considera el promedio de estas disimilitudes como la distancia entre los dos grupos.

Al final se coerciona el objeto clustering a dendograma.

tbb\_dendo <- read\_csv(here("R\_Bursera", "B\_data", "8\_B\_biflora\_R.csv"))%>%  
 select(-fid, -gbifID, -species, -month, -year) %>%   
 scale( ) %>%   
 as\_tibble() %>%   
 dist( method = "euclidean") %>%   
 hclust(method = "average") %>%   
 as.dendrogram( )

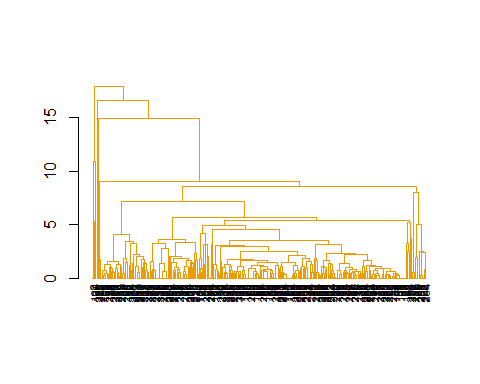
## Rows: 301 Columns: 27  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): species  
## dbl (26): fid, gbifID, decimalLat, decimalLon, month, year, 1\_Annual\_Mean\_Te...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

## inicia el proceso de exportacion del plot

jpeg(file = here("R\_Bursera", "B\_output", "B8\_dendo.jpeg"), width = 1000, height = 500)

Visualización de un dendograma reduciendo el tamaño de las etiquetas, “labels\_cex”, a un 60%. Las ramas del dendograma se colorea de acuerdo al número de grupos especificado con “k”, el color en “orange2”

tbb\_dendo %>% set( what = "labels\_cex", value = 0.6) %>%   
 set( "branches\_k\_color", k = 1, value = c( "orange2")) %>%  
 plot()



#————Similitud——- #altura máxima del dendo para calcular dendro #asignar un objeto con la altura máxima para la visualización y el cálculo de abline altura\_max <- as.numeric(c(28.12245))

#Etiqueta del eje X para mejor visualización

#Superponer un nuevo eje en porcentaje en el lado derecho axis(side = 4, # 4 = lado derecho del gráfico at = seq(0, altura\_max, length.out = 11), # Posiciones en el eje original labels = paste0(seq(100, 0, by = -10), “%”)) # Etiquetas en porcentaje

#calcular el 75% h <- ((40\*altura\_max)/100) h dendro <- abline(h= h) #genera una linea en la altura especificada

#temina el proceso de exportacion del plot dev.off()

#———Poda del dendograma (separación de clústers)——–

#decidir en base a abline el valor k (numero de clusters) dendro <- as.dendrogram( clustering) #coerciona el objeto clusterin a dendograma clusters <-cutree(dendro, k= 7, order\_clusters\_as\_data = FALSE) #crea un objeto tipo integrer y lo corta en el numero K de grupos

plot(color\_branches(dendro, k=7),leaflab=“none”) #genera un dendograma con color con el numero K de clusters indicado

#genera un objeto de tipo data frame usando los clusters “cutree”, se declaran las columnas o vectores clusters\_df <- data.frame(ID = names(clusters), cluster = clusters)

#crear SÓLO el número de clústers encontrado en abline cluster1 <- filter(clusters\_df, cluster == 1)ID cluster3 <- filter(clusters\_df, cluster == 3)ID cluster5 <- filter(clusters\_df, cluster == 5)ID cluster7 <- filter(clusters\_df, cluster == 7)ID cluster9 <- filter(clusters\_df, cluster == 9)ID

table(clusters) #imprime el numero de obserbaciones que integran cada cluster

#preparación de datos para el archivo txt con filas a descartar (eliminados\_\_dendo\_I1)/ #genera una lista de los integrantes de un determinado cluster cat(as.character(cluster1[1:42]), quote=FALSE,sep=“”) cat(as.character(cluster2[1:2]), quote=FALSE,sep=“”) cat(as.character(cluster3[1:17]), quote=FALSE,sep=“”) cat(as.character(cluster4[1:1]), quote=FALSE,sep=“”) cat(as.character(cluster5[1:5]), quote=FALSE,sep=“”) cat(as.character(cluster6[1:9]), quote=FALSE,sep=“”) cat(as.character(cluster7[1:9]), quote=FALSE,sep=“”) cat(as.character(cluster8[1:2]), quote=FALSE,sep=“”) cat(as.character(cluster9[1:2]), quote=FALSE,sep=“”) cat(as.character(cluster10[1:2]), quote=FALSE,sep=“”)

#OPCIONAL———— generar un nuevo dendograma con los subárboles a la altura dada———- sub.trees <- cut(dendro, h = 15) sub.treeslower[[2]] clusterLow %>% set( what = “labels\_cex”, value = 0.6) %>% set( “branches\_k\_color”, k = 1, value = c( “orange2”)) %>% plot() #—————–

#————-creacion de nueva base de datos con filas eliminadas en el filtro dendo————————————-

#crea un objeto leyendo (read\_csv) la base de datos csv obtenida con la limpieza en openrefine y qgis, usa la paqueteria “here” para ubicar el archivo

I\_spthin <- read\_csv(here(“R\_Icterus”, “I\_data”, “1\_I\_abeillei\_R.csv”)) numeros\_eliminar <- scan(here(“R\_Icterus”, “I\_data”, “eliminados\_\_dendo\_I1.txt”))

I\_thin <- I\_spthin[-numeros\_eliminar, ]

#———–DENDOGRAMA print cuántos registros se conservaron—————-

# Leer el archivos y contar filas

#lee la base de datos inicial y despues cuenta las filas sin tener encuenta los encabezados datos\_ini\_csv <- read\_csv(here(“R\_Icterus”, “I\_data”, “1\_I\_abeillei\_R.csv”)) datos\_inicial <- nrow(datos\_ini\_csv)

#lee el archivo txt generado anteriormente, con las filas a eliminar, y despues cuenta las filas. Se especifica que no tiene encabezados y debe leer desde la primera linea datos\_txt <- read.table(here(“R\_Icterus”, “I\_data”, “eliminados\_\_dendo\_I1.txt”), header = FALSE) datos\_elim <- nrow(datos\_txt)

#hace una resta de los objetos para calcular el n’umero de datos posteriores al dendograma dendo\_datos\_final <- datos\_inicial - datos\_elim

print(paste0(“Dendograma/ número inicial:”, datos\_inicial, “, eliminados:”, datos\_elim, “, Número final:”, dendo\_datos\_final))

#–OPCIONAL—————–codigo para crear un dendo con la nueva base de datos———————– #crea un objeto llamado “limpio” donde se descartan los vectores (columnas) que no aportan información útil para la creación del clúster limpio <- I1\_thin %>% select(-shapeid, -partid, -x, -y, -fid, -gbifID, -species, -decimalLat, -decimalLon, -month, -year)

#crea un objeto llamado “escalado” escalando el objeto limpio para poder realizar el càlculo de las distancias y coercionándolo a tibble escalado <- scale( x = limpio) %>% as\_tibble()

#crea yna matriz de distancias por distancias euclidianas distancia <- dist( x = escalado, method = “euclidean”)

#crea un clùster\_–Agrupamiento por ligamiento medio o promedio: calcula todas las disimilitudes por pares entre los elementos del grupo 1 y los elementos del grupo 2, y considera el promedio de estas disimilitudes como la distancia entre los dos grupos. clustering <- hclust( d= distancia, method = “average”)

dendro <- as.dendrogram( clustering) #coerciona el objeto clustering a dendograma

dendro %>% set( what = “labels\_cex”, value = 0.6) %>% set( “branches\_k\_color”, k = 1, value = c( “orange2”)) %>% plot() #crea un plot con las caracteristicas especidficadas #——————————————

#————-SpThin————————————-

#install.packages(“spThin”)—-este paquete contiene un conjunto de funciones para realizar el adelgazamiento espacial de registros de ocurrencia lo que busca equilibrar la densidad de registros library(spThin)

head(“I\_thin”) plot(I\_thindecimalLat, xlab = “Longitud”, #etiquetas ejes ylab = “Latitud”, sub = “Icterus abeillei”, #subtitulo cex.sub = 0.5, #size sub (50%) font.sub = 3) #fuente sub cursiva (negrita 2/ negrita-cursiva 4)

thin( loc.data = I\_thin, #base de datos a procesar, se usa el objeto “I\_thin” que contiene la base resultante del filtro dendo lat.col = “x”, long.col = “y”, spec.col = “shapeid”, thin.par = 10, reps = 100, #thin.par: distancia mínima entre puntos en km. reps = repeticiones del proceso de adelgazamiento. locs.thinned.list.return = TRUE, #TRUE se genera una lista de cada repetición realizada write.files = TRUE, #TRUE genera archivos con extensión csv. write.log.file = TRUE, out.dir = here(“R\_Icterus”, “I\_data”, “output\_I1\_thin” ))

#———–ADELGAZAMIENTO print cuántos registros se conservaron—————-

# Leer el archivos y contar filas

#se toma el objeto resultado del conteo filtro dendograma datos\_inicial <- dendo\_datos\_final

#lee el archivo generado en spthin con los datos finales y despues cuenta las filas sin tomar en cuenta encabezados datos\_fin\_csv <- read\_csv(here(“R\_Icterus”, “I\_data”, “output\_I1\_thin”,“thinned\_data\_thin1.csv”)) spthin\_datos\_final <- nrow(datos\_fin\_csv)

#hace una resta de los objetos para calcular el n’umero de datos posteriores al adelgazamiento datos\_elim <- datos\_inicial - spthin\_datos\_final

print(paste0(“spThin/ Número inicial:”, datos\_inicial, “, eliminados:”, datos\_elim, “, Número final:”, spthin\_datos\_final))

#————————write id de observaciones conservadas después del adelazamiento————— #lectura de registros iniciales con id thinned\_data\_thin1 <- read\_csv(here(“R\_Icterus”, “I\_data”,“output\_I1\_thin”, “thinned\_data\_thin1.csv”)) #lectura de resultados de adelgazamiento inicial <- read\_csv(here(“R\_Icterus”, “I\_data”, “1\_I\_abeillei\_R.csv”))

# join usando las coordenadas y la especie como clave

id\_spthin <- thinned\_data\_thin1 %>% left\_join(inicial, by = c(“species” = “species”, # “thinned\_data\_thin1” = “inicial” “x” = “decimalLon” , “y” = “decimalLat” ))

#Verificar NAs para buscar errores de no coincidencia which(is.na(id\_spthin$fid))

#arreglar formato colnames(id\_spthin)[2] <- “decimalLat” colnames(id\_spthin)[3] <- “decimalLon” id\_spthin <- id\_spthin %>% relocate(fid)

#guardar un csv write\_csv(id\_spthin, “R\_Icterus/I\_data/output\_I1\_thin/id\_spthin.csv”)

#————————puntos adelgazados————————-

sthin <- read\_csv(here(“R\_Icterus”, “I\_data”, “output\_I1\_thin”, “thinned\_data\_thin1.csv”)) points(sthiny, col = “orange”, pch = 20)

library(tidyverse) library(“pacman”) p\_load(“vroom”, “dplyr”, “dendextend”) library(readr) library(cluster) library(here) library(spThin) library(readr) library(“colorspace”) #———-Generación de dendograma———— I10\_dominicensis <- read\_csv(here(“R\_Icterus”, “I\_data”, “10\_I\_dominicensis\_R.csv”)) limpio <- I10\_dominicensis %>% select(-fid, -gbifID, -species, -month, -year) escalado <- scale( x = limpio) %>% as\_tibble() distancia <- dist( x = escalado, method = “euclidean”) clustering <- hclust( d= distancia, method = “average”) dendo <- as\_dendrogram( clustering)

#inicia el proceso de exportacion del plot png(file = “C:/Users/Yas/OneDrive/Maestría/proyecto/QGis/layers/Icterus\_proyect/R\_Icterus/I\_data/graphics/I\_10dendo.png”, width = 5000, height = 2000)

dendo %>% set( what = “labels\_cex”, value = 0.3) %>% set( “branches\_k\_color”, k = 1, value = c( “orange2”)) %>% plot()

#————Similitud——- #altura máxima del dendo para calcular dendo #asignar un objeto con la altura máxima para la visualización y el cálculo de abline altura\_max <- as.numeric(c(32.45111))

#Etiqueta del eje X para mejor visualización————–

#Superponer un nuevo eje en porcentaje en el lado derecho axis(side = 4, # 4 = lado derecho del gráfico at = seq(0, altura\_max, length.out = 11), # Posiciones en el eje original labels = paste0(seq(100, 0, by = -10), “%”)) # Etiquetas en porcentaje

#temina el proceso de exportacion del plot dev.off()

#calcular el 75% h <- ((25\*altura\_max)/100) h dendo <- abline(h= h)

#———Poda del dendograma (separación de clústers)——–

#decidir en base a abline el valor k (numero de clusters) dendo <- as.dendrogram( clustering) clusters <-cutree(dendo, k= 5, order\_clusters\_as\_data = FALSE)

#Opcional —–dendo <-rect.dendrogram(dendo, k = 21)

#opcional plot(color\_branches(dendo, k=5),leaflab=“none”) #genera un dendograma con color con el numero K de clusters indicado

#crear SÓLO el número de clústers encontrado en abline clusters\_df <- data.frame(ID = names(clusters), cluster = clusters) cluster1 <- filter(clusters\_df, cluster == 1)ID cluster3 <- filter(clusters\_df, cluster == 3)ID cluster5 <- filter(clusters\_df, cluster == 5)$ID

table(clusters) #imprime el numero de obserbaciones que integran cada cluster

#preparación de datos para el archivo txt con filas a descartar (eliminados\_\_dendo\_I1) cat(as.character(cluster1[1]), quote=FALSE,sep=“”) cat(as.character(cluster2[1]), quote=FALSE,sep=“”) cat(as.character(cluster3[1:17]), quote=FALSE,sep=“”) cat(as.character(cluster4[1:33]), quote=FALSE,sep=“”)

#–OPCIONAL—————–codigo para crear un dendo con la nueva base de datos———————– sub.trees <- cut(dendo, h = 15) sub.treeslower[[2]] clusterLow %>% set( what = “labels\_cex”, value = 0.6) %>% set( “branches\_k\_color”, k = 1, value = c( “orange2”)) %>% plot() #-\*\*\*\*\*\*\*————————————–

#————-creacion de nueva base de datos con filas eliminadas en el filtro dendo————————————-

I\_spthin <- read\_csv(here(“R\_Icterus”, “I\_data”, “10\_I\_dominicensis\_R.csv”)) numeros\_eliminar <- scan(here(“R\_Icterus”, “I\_data”, “eliminados\_\_dendo\_I10.txt”))

I\_thin <- I\_spthin[-numeros\_eliminar, ] I\_thin

#———–DENDOGRAMA print cuántos registros se conservaron—————-

# Leer el archivos y contar filas

datos\_ini\_csv <- read\_csv(here(“R\_Icterus”, “I\_data”, “10\_I\_dominicensis\_R.csv”)) datos\_inicial <- nrow(datos\_ini\_csv)

datos\_txt <- read.table(here(“R\_Icterus”, “I\_data”, “eliminados\_\_dendo\_I10.txt”), header = FALSE) datos\_elim <- nrow(datos\_txt)

dendo\_datos\_final <- datos\_inicial - datos\_elim

print(paste0(“Dendograma/ número inicial:”, datos\_inicial, “, eliminados:”, datos\_elim, “, Número final:”, dendo\_datos\_final))

#————-SpThin————————————-

png(file = “C:/Users/Yas/OneDrive/Maestría/proyecto/QGis/layers/Icterus\_proyect/R\_Icterus/I\_data/graphics/I\_10spthin.png”, width = 666, height = 740)

head(“I\_thin”) plot(I\_thindecimalLat, xlab = “Longitud”, ylab = “Latitud”, sub = “Icterus dominicensis”, cex.sub = 0.5, font.sub = 3)

thin( loc.data = I\_thin, lat.col = “decimalLat”, long.col = “decimalLon”, spec.col = “species”, thin.par = 10, reps = 100, locs.thinned.list.return = TRUE, write.files = TRUE, write.log.file = TRUE, out.dir = here(“R\_Icterus”, “I\_data”, “output\_I10\_thin” ))

#———–ADELGAZAMIENTO print cuántos registros se conservaron—————-

# Leer el archivos y contar filas

datos\_inicial <- dendo\_datos\_final

datos\_fin\_csv <- read\_csv(here(“R\_Icterus”, “I\_data”, “output\_I10\_thin”,“thinned\_data\_thin1.csv”)) spthin\_datos\_final <- nrow(datos\_fin\_csv)

datos\_elim <- datos\_inicial - spthin\_datos\_final

print(paste0(“spThin/ Número inicial:”, datos\_inicial, “, eliminados:”, datos\_elim, “, Número final:”, spthin\_datos\_final))

#————————write id de observaciones conservadas después del adelazamiento————— #lectura de registros iniciales con id thinned\_data\_thin1 <- read\_csv(here(“R\_Icterus”, “I\_data”,“output\_I10\_thin”, “thinned\_data\_thin1.csv”)) #lectura de resultados de adelgazamiento inicial <- read\_csv(here(“R\_Icterus”, “I\_data”, “10\_I\_dominicensis\_R.csv”))

# join usando las coordenadas y la especie como clave

id\_spthin <- thinned\_data\_thin1 %>% left\_join(inicial, by = c(“species” = “species”, “decimalLon” = “decimalLon”, “decimalLat” = “decimalLat” )) # “thinned\_data\_thin1” = “inicial”

#Verificar NAs para buscar errores de no coincidencia which(is.na(id\_spthin$fid))

#arreglar formato colnames(id\_spthin)[2] <- “decimalLat” colnames(id\_spthin)[3] <- “decimalLon” id\_spthin <- id\_spthin %>% relocate(fid)

#guardar un csv write\_csv(id\_spthin, “R\_Icterus/I\_data/output\_I10\_thin/id\_spthin.csv”)

#————————puntos adelgazados————————-

sthin <- read\_csv(here(“R\_Icterus”, “I\_data”, “output\_I10\_thin”, “thinned\_data\_thin1.csv”)) points(sthindecimalLat, col = “orange”, pch = 20) dev.off()

## 