

Helechos a mansalva

Luis Daniel Fernández Montesinos

2025-01-09

Preliminares

- Cargar paquetes.

```
library(vegan)
library(sf)
library(tidyverse)
library(tmap)
library(kableExtra)
library(broom)
library(cluster)
library(gclus)
library(pvclust)
library(foreach)
library(leaps)
library(caret)
library(RColorBrewer)
library(indicspecies)
library(dendextend)
library(adespatial)
library(SpadeR)
library(iNEXT)
library(GGally)
library(vegetarian)
gh_content <- 'https://raw.githubusercontent.com/'
gh_zonal_stats <- paste0(gh_content,
                          'geofis/zonal-statistics/0b2e95aaee87bf326cf132d28f4bd15220bb4ec7/out/')
repo_analisis <- 'biogeografia-master/scripts-de-analisis-BCI/master'
repo_sem202202 <- 'biogeografia-202202/material-de-apoyo/master/practicas/'
devtools::source_url(paste0(gh_content, repo_analisis, '/biodata/funciones.R'))
devtools::source_url(paste0(gh_content, repo_sem202202, 'train.R'))
devtools::source_url(paste0(gh_content, repo_sem202202, 'funciones.R'))
source('R/funciones.R')
umbral_alfa <- 0.05
```

Análisis exploratorio de datos (AED)

- Cargar datos.

```
datos_orig <- read_csv('RCEV.csv')
grupos_seleccionados <- rep(sort(unique(datos_orig$Localidad)), each = 3)
datos <- datos_orig %>%
  mutate(id = paste0(Localidad, '-', Transecto)) %>%
  select(-Localidad, -Transecto) %>%
```

```

group_by(Especies, Abundancia, id) %>%
distinct() # Para eliminar registros redundantes
transect_coords <- read_csv('transect_coordinates.csv') %>%
mutate(id = paste0(Location, '-', `Transect number`)) %>%
select(-Location, -`Transect number`, -Elevation) %>%
pivot_longer(-id, names_to = 'Coordinate type') %>%
separate(col = value, into = c('Latitude', 'Longitude'), convert = T, sep = ',')
transect_coords_sf <- transect_coords %>%
st_as_sf(coords = c('Longitude', 'Latitude'), crs = 4326, remove = F)
# transect_coords_sf %>% st_write('transect_coordinates.kml')

```

- Generar matriz de comunidad.

```

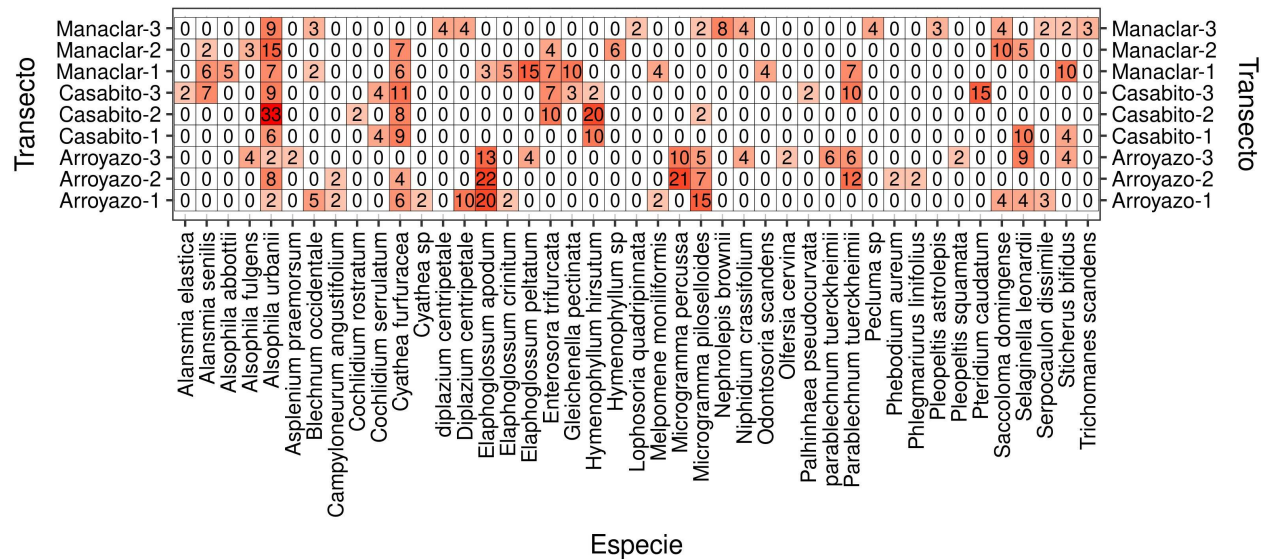
mc <- datos %>%
pivot_wider(names_from = Especies, values_from = Abundancia, values_fill = 0) %>%
column_to_rownames('id')

mc_pooled <- mc %>%
rownames_to_column('sitio') %>%
mutate(sitio = gsub('-', '*', '', sitio)) %>%
group_by(sitio) %>%
summarise(across(is.numeric, ~ sum(.x))) %>%
column_to_rownames('sitio')

```

Gráfico de mosaico

```
crear_grafico_mosaico_de_mc(mc)
```



Preferencia

```

set.seed(9999)
indval <- multipatt(
  mc,
  grupos_seleccionados,
  func = "IndVal.g",

```

```

max.order = 1,
control = how(nperm = 999))
summary(indval)

```

```

##
## Multilevel pattern analysis
## -----
##
## Association function: IndVal.g
## Significance level (alpha): 0.05
##
## Total number of species: 43
## Selected number of species: 4
## Number of species associated to 1 group: 4
## Number of species associated to 2 groups: 0
##
## List of species associated to each combination:
##
## Group Arroyazo #sps. 2
##
##               stat p.value
## Elaphoglossum apodum    0.974  0.026 *
## Microgramma piloselloides 0.933  0.026 *
##
## Group Casabito #sps. 2
##
##               stat p.value
## Hymenophyllum hirsutum 1.000  0.028 *
## Cyathea furfuracea    0.741  0.028 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

phi <- multipatt(
  mc,
  grupos_seleccionados,
  func = "r.g",
  max.order = 1,
  control = how(nperm = 999))
summary(phi)

```

```

##
## Multilevel pattern analysis
## -----
##
## Association function: r.g
## Significance level (alpha): 0.05
##
## Total number of species: 43
## Selected number of species: 4
## Number of species associated to 1 group: 4
## Number of species associated to 2 groups: 0
##
## List of species associated to each combination:
##
## Group Arroyazo #sps. 2
##
##               stat p.value

```

```
## Elaphoglossum apodum      0.961  0.041 *
## Microgramma piloselloides 0.833  0.041 *
##
## Group Casabito #sps. 2
##                      stat p.value
## Hymenophyllum hirsutum 0.764  0.033 *
## Cyathea furfuracea     0.728  0.033 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Índices

```
indices <- alpha_div(mc) %>%
  mutate(sitio = rownames(.)) %>%
  relocate(sitio, .before = everything())
indices %>% estilo_kable()
```

Table 1:

sitio	N0	H	Hb2	N1	N1b2	N2	J	E10	E20
Arroyazo-1	14	2.46	3.55	11.74	11.74	10.11	0.93	0.84	0.72
Arroyazo-2	9	1.86	2.68	6.43	6.43	5.29	0.85	0.71	0.59
Arroyazo-3	13	2.22	3.20	9.19	9.19	7.00	0.86	0.71	0.54
Casabito-1	11	2.19	3.16	8.96	8.96	7.83	0.91	0.81	0.71
Casabito-2	6	1.41	2.04	4.11	4.11	3.39	0.79	0.69	0.56
Casabito-3	6	1.72	2.48	5.60	5.60	5.30	0.96	0.93	0.88
Manaclar-1	14	2.52	3.63	12.37	12.37	10.88	0.95	0.88	0.78
Manaclar-2	8	1.91	2.75	6.73	6.73	5.83	0.92	0.84	0.73
Manaclar-3	14	2.52	3.64	12.46	12.46	11.21	0.96	0.89	0.80

```
indices_pooled <- alpha_div(mc_pooled) %>%
  mutate(sitio = rownames(.)) %>%
  relocate(sitio, .before = everything())
indices_pooled %>% estilo_kable()
```

Table 2:

sitio	N0	H	Hb2	N1	N1b2	N2	J	E10	E20
Arroyazo	25	2.64	3.81	14.01	14.01	9.20	0.82	0.56	0.37
Casabito	15	2.25	3.25	9.48	9.48	7.25	0.83	0.63	0.48
Manaclar	28	3.06	4.42	21.43	21.43	16.35	0.92	0.77	0.58

“Complejidad de muestra” y curva de acumulación

- “Complejidad de muestra”, estimadores tradicionales

```
riqueza_estimaciones <- data.frame(estimateR(mc)) %>%
  rownames_to_column('Estimador') %>%
  pivot_longer(-Estimador, names_to = 'Sitio', values_to = 'Estimado') %>%
  pivot_wider(names_from = Estimador, values_from = Estimado) %>%
```

```
select(
  Sitio,
  `Riqueza observada` = S.obs,
  `Estimación por Chao1` = S.chao1,
  `Estimación por ACE` = S.ACE)
riqueza_estimaciones %>% estilo_kable(alinear = 'lrr')
```

Table 3:

Sitio	Riqueza observada	Estimación por Chao1	Estimación por ACE
Arroyazo.1	14	14	14
Arroyazo.2	9	9	9
Arroyazo.3	13	13	13
Casabito.1	11	11	11
Casabito.2	6	6	6
Casabito.3	6	6	6
Manaclar.1	14	14	14
Manaclar.2	8	8	8
Manaclar.3	14	14	14

```
riqueza_estimaciones_pooled <- data.frame(estimateR(mc_pooled)) %>%
  rownames_to_column('Estimador') %>%
  pivot_longer(-Estimador, names_to = 'Sitio', values_to = 'Estimado') %>%
  pivot_wider(names_from = Estimador, values_from = Estimado) %>%
  select(
    Sitio,
    `Riqueza observada` = S.obs,
    `Estimación por Chao1` = S.chao1,
    `Estimación por ACE` = S.ACE)
riqueza_estimaciones_pooled %>% estilo_kable(alinear = 'lrr')
```

Table 4:

Sitio	Riqueza observada	Estimación por Chao1	Estimación por ACE
Arroyazo	25	25	25
Casabito	15	15	15
Manaclar	28	28	28

- Estimador Chao mejorado

```
df_spader <- data.frame(V1 = as.integer(sort(colSums(mc), decreasing = T)))
ChaoSpecies(df_spader, datatype = 'abundance',
  k = min(df_spader$V1), conf=0.95)
```

```
## Warning: In this case, it can't estimate the variance of 2nd-order-jackknife estimation
```

```
##
```

```
## (1) BASIC DATA INFORMATION:
```

```
##
```

```
##
```

```
## Variable Value
```

```
## Sample size n 617
```

```
##      Number of observed species          D      43
##      Coverage estimate for entire dataset    C      1
##      CV for entire dataset                  CV 1.217
##      Cut-off point                          k      2
##
##
##                                     Variable Value
##      Number of observed individuals for rare group    n_rare    20
##      Number of observed species for rare group        D_rare    10
##      Estimate of the sample coverage for rare group    C_rare     1
##      Estimate of CV for rare group in ACE              CV_rare     0
##      Estimate of CV1 for rare group in ACE-1          CV1_rare    0
##      Number of observed individuals for abundant group n_abun   597
##      Number of observed species for abundant group    D_abun    33
```

```
## NULL
```

(2) SPECIES RICHNESS ESTIMATORS TABLE:

```
##                                     Estimate  s.e. 95%Lower 95%Upper
##      Homogeneous Model                43 1.158      43 46.963
##      Homogeneous (MLE)                43 0.005      43 43.015
##      Chao1 (Chao, 1984)               43 1.158      43 46.963
##      Chao1-bc                         43 1.158      43 46.963
##      iChao1 (Chiu et al. 2014)        43 1.158      43 46.963
##      ACE (Chao & Lee, 1992)           43 1.158      43 46.963
##      ACE-1 (Chao & Lee, 1992)         43 1.158      43 46.963
##      1st order jackknife              43 1.158      43 46.963
##      2nd order jackknife              43 1.158      43 46.963
```

(3) DESCRIPTION OF ESTIMATORS/MODELS:

```
## Homogeneous Model: This model assumes that all species have the same incidence or detection probability.
```

```
## Chao2 (Chao, 1987): This approach uses the frequencies of uniques and duplicates to estimate the number of species.
```

```
## Chao2-bc: A bias-corrected form for the Chao2 estimator; see Chao (2005).
```

```
## iChao2: An improved Chao2 estimator; see Chiu et al. (2014).
```

```
## ICE (Incidence-based Coverage Estimator): A non-parametric estimator originally proposed by Lee and Chao (2003).
```

```
## ICE-1: A modified ICE for highly-heterogeneous cases.
```

```
## 1st order jackknife: It uses the frequency of uniques to estimate the number of undetected species; see Chao (1987).
```

```
## 2nd order jackknife: It uses the frequencies of uniques and duplicates to estimate the number of undetected species; see Chao (1987).
```

```
## 95% Confidence interval: A log-transformation is used for all estimators so that the lower bound of the confidence interval is not zero.
```

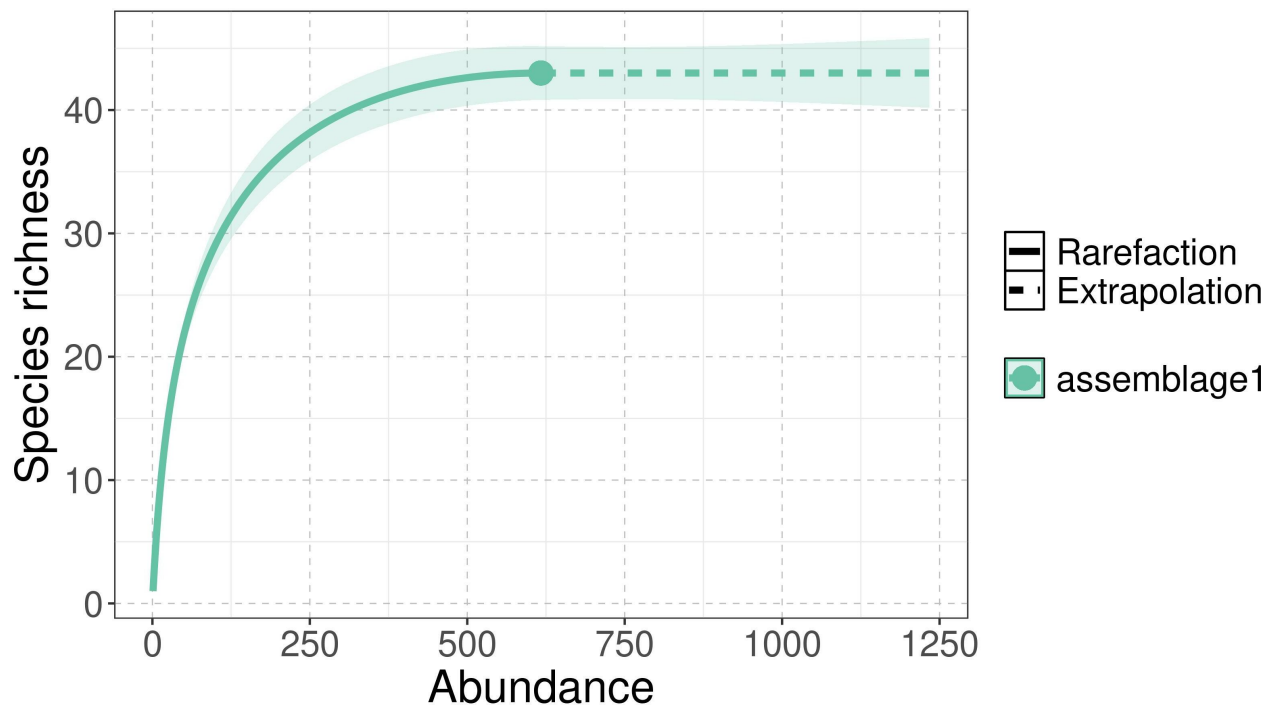
- Curva de acumulación

```
mc_general <- mc %>%
  summarise_all(sum) %>%
```

```

# mutate(N = nrow(mc)) %>%
# relocate(N, .before = 1) %>%
data.frame
nasin_raref <- iNEXT::iNEXT(
  x = t(mc_general),
  q=0,
  knots = 2000,
  datatype = 'abundance')
acumulacion_especies <- iNEXT::ggiNEXT(nasin_raref, type=1) +
  theme_bw() +
  theme(
    text = element_text(size = 20),
    panel.background = element_rect(fill = 'white', colour = 'black'),
    panel.grid.major = element_line(colour = "grey", linetype = "dashed", size = 0.25)
  ) +
  ylab('Species richness') +
  xlab('Abundance') +
  scale_y_continuous(breaks = seq(0, 80, length.out = 9)) +
  scale_color_manual(values = brewer.pal(8, 'Set2')) +
  scale_fill_manual(values = brewer.pal(8, 'Set2'))
acumulacion_especies

```



- Curva de acumulación por tipo de bosque

```

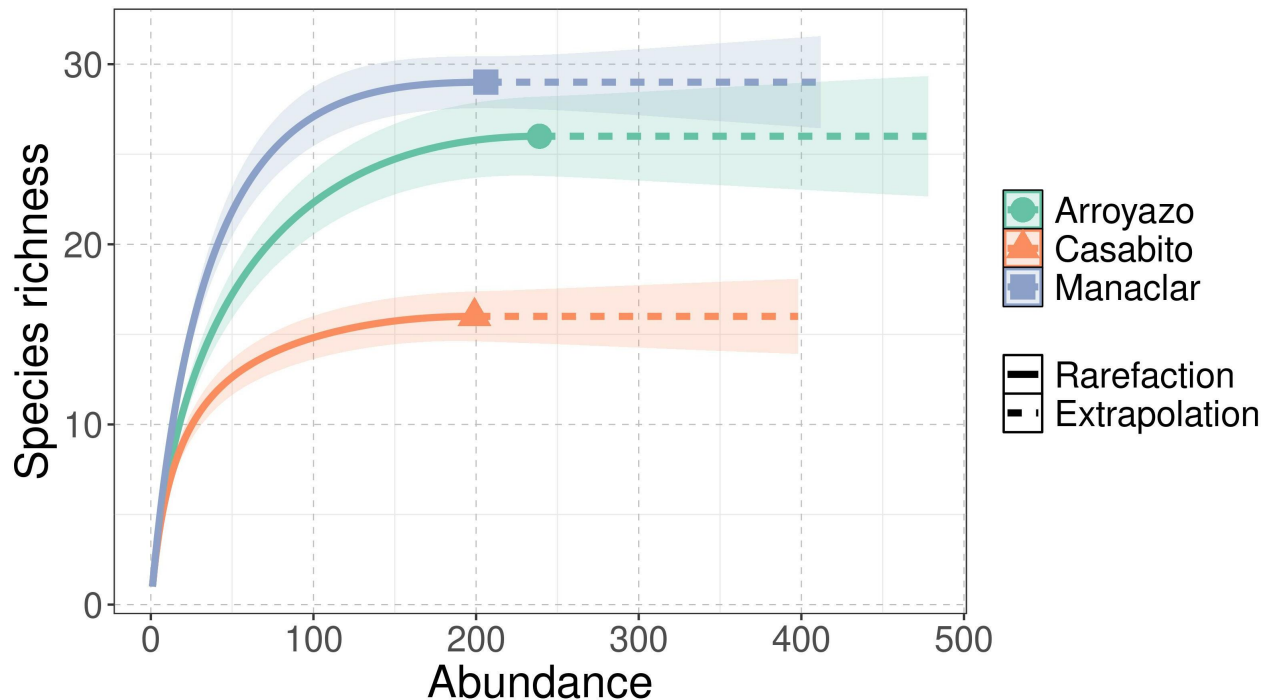
mc_grupos <- mc %>%
  mutate(g = grupos_seleccionados) %>%
  group_by(g) %>%
  summarise_all(sum) %>%
  select(-g) %>%
  mutate(N = nrow(mc)) %>%
  relocate(N, .before = 1) %>%
data.frame

```

```

nasin_raref_general <- iNEXT::iNEXT(
  x = t(mc_grupos),
  q=0,
  knots = 400,
  datatype = 'abundance')
acumulacion_especies_grupos <- iNEXT::ggiNEXT(nasin_raref_general, type=1) +
  theme_bw() +
  theme(
    text = element_text(size = 20),
    panel.background = element_rect(fill = 'white', colour = 'black'),
    panel.grid.major = element_line(colour = "grey", linetype = "dashed", size = 0.25)
  ) +
  ylab('Species richness') +
  xlab('Abundance') +
  scale_y_continuous(breaks = seq(0, 80, length.out = 9)) +
  scale_color_manual(values = brewer.pal(8, 'Set2'), labels = unique(grupos_seleccionados)) +
  scale_fill_manual(values = brewer.pal(8, 'Set2'), labels = unique(grupos_seleccionados)) +
  scale_shape_manual(values = c(16, 17, 15), labels = unique(grupos_seleccionados))
acumulacion_especies_grupos

```



Asociación con variables ambientales

Retomando idea compartida por correo:

...se podría explorar, aun siendo pocos transectos (ya sé que en terreno tres transectos por formación es mucho, pero estadísticamente es poco), asociación con variables geomorfológicas, distancia a cursos fluviales, litología, cubierta y altura de dosel, elevación y variables climáticas. Lógicamente, para esto, sería necesario disponer de las coordenadas de los transectos.

- Estadística zonal

Scripts heredados

Hill.R

```
# Cargar los datos desde el archivo RCEV.csv
ruta_archivo <- "C:/Users/jazmi/OneDrive/Documents/tesis-daniel/RCEV.csv"
datos <- read.csv(ruta_archivo)
library(dplyr)
library(vegan)
library(tidyr)
library(ggplot2)

# Suponiendo que tus datos están en un data frame llamado 'datos'
# Aquí hay un ejemplo de cómo pueden verse tus datos
# datos <- data.frame(
#   Localidad = c("Loc1", "Loc1", "Loc2", "Loc2"),
#   Transecto = c("Tran1", "Tran1", "Tran2", "Tran2"),
#   Especies = c("Especie1", "Especie2", "Especie1", "Especie2"),
#   Abundancia = c(10, 15, 5, 20)
# )

# Calcular los Números de Hill
calcular_hill <- function(data) {
  # Crear una matriz de abundancias por especies
  matriz_abundancias <- data %>%
    group_by(Localidad, Especies) %>%
    summarize(Abundancia = sum(Abundancia), .groups = 'drop') %>%
    spread(Especies, Abundancia, fill = 0)

  # Eliminar la columna de Localidad para la matriz de abundancia
  localidades <- matriz_abundancias$Localidad
  matriz_abundancias <- matriz_abundancias %>% select(-Localidad)

  # Calcular los índices de diversidad
  H0 <- rowSums(matriz_abundancias > 0) # Riqueza de especies
  H1 <- exp(diversity(matriz_abundancias, index = "shannon")) # Exponencial del índice de Shannon
  H2 <- 1 / diversity(matriz_abundancias, index = "simpson") # Inverso del índice de Simpson

  return(data.frame(Localidad = localidades, H0 = H0, H1 = H1, H2 = H2))
}

# Calcular los números de Hill para tus datos
hill_results <- calcular_hill(datos)

# Imprimir resultados
print(hill_results)

# Transformar los resultados para graficar
hill_df_long <- gather(hill_results, key = "Indice", value = "Valor", -Localidad)

# Graficar los resultados
ggplot(hill_df_long, aes(x = Localidad, y = Valor, fill = Indice)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_minimal() +
  labs(title = "Números de Hill", x = "Localidad", y = "Valor del Índice") +
```

```
scale_fill_manual(values = c("H0" = "blue", "H1" = "red", "H2" = "green")) +
theme(plot.title = element_text(hjust = 0.5))
```

similitud.R

```
# Cargar los datos desde el archivo RCEV.csv
ruta_archivo <- "C:/Users/jazmi/OneDrive/Documents/tesis-daniel/RCEV.csv"
datos <- read.csv(ruta_archivo)
# Instalar y cargar las librerías necesarias

library(heatmaply)

# Definir una función para calcular la similitud de Jaccard entre tres localidades específicas
calcular_similitud_jaccard_tres_localidades <- function(datos, localidad1, localidad2, localidad3) {
  # Obtener los datos de las tres localidades
  datos_localidades <- datos[datos$Localidad %in% c(localidad1, localidad2, localidad3), ]

  # Obtener las especies únicas en las tres localidades
  especies <- unique(datos_localidades$Especies)

  # Crear una matriz binaria de presencia/ausencia para las especies en las tres localidades
  datos_binarios <- matrix(0, nrow = 3, ncol = length(especies))
  rownames(datos_binarios) <- c(localidad1, localidad2, localidad3)
  colnames(datos_binarios) <- especies

  for (i in 1:nrow(datos_localidades)) {
    localidad <- datos_localidades[i, "Localidad"]
    especie <- datos_localidades[i, "Especies"]
    abundancia <- datos_localidades[i, "Abundancia"]
    datos_binarios[localidad, especie] <- ifelse(abundancia > 0, 1, 0)
  }

  # Calcular la similitud de Jaccard entre las tres localidades
  similitud_jaccard <- matrix(NA, nrow = 3, ncol = 3)
  for (i in 1:2) {
    for (j in (i+1):3) {
      numerador <- sum(datos_binarios[i,] & datos_binarios[j,])
      denominador <- sum(datos_binarios[i,] | datos_binarios[j,])
      similitud_jaccard[i,j] <- numerador / denominador
      similitud_jaccard[j,i] <- similitud_jaccard[i,j]
    }
  }

  return(similitud_jaccard)
}

# Calcular la similitud de Jaccard entre las tres localidades específicas
similitud_jaccard_tres_localidades <- calcular_similitud_jaccard_tres_localidades(datos, "Casabito", "M"
```

```
# Crear el mapa de calor utilizando heatmaply
heatmaply(similitud_jaccard_tres_localidades,
  symmetric = TRUE,
  labRow = c("Casabito", "Manaclar", "Arroyazo"),
```

```

labCol = c("Casabito", "Manaclar", "Arroyazo"),
main = "Mapa de Calor de Similitud de Jaccard entre Localidades",
col = c("red", "orange", "green"))

#####
# Cargar los datos desde el archivo RCEV.csv
ruta_archivo <- "C:/Users/jazmi/OneDrive/Documents/tesis-daniel/RCEV.csv"
datos <- read.csv(ruta_archivo)

# Instalar y cargar las librerías necesarias
if (!requireNamespace("heatmaply", quietly = TRUE)) {
  install.packages("heatmaply")
}
library(heatmaply)

# Definir una función para calcular la similitud de Jaccard entre tres localidades específicas
calcular_similitud_jaccard_tres_localidades <- function(datos, localidad1, localidad2, localidad3) {
  # Obtener los datos de las tres localidades
  datos_localidades <- datos[datos$Localidad %in% c(localidad1, localidad2, localidad3), ]

  # Obtener las especies únicas en las tres localidades
  especies <- unique(datos_localidades$Especies)

  # Crear una matriz binaria de presencia/ausencia para las especies en las tres localidades
  datos_binarios <- matrix(0, nrow = 3, ncol = length(especies))
  rownames(datos_binarios) <- c(localidad1, localidad2, localidad3)
  colnames(datos_binarios) <- especies

  for (i in 1:nrow(datos_localidades)) {
    localidad <- datos_localidades[i, "Localidad"]
    especie <- datos_localidades[i, "Especies"]
    abundancia <- datos_localidades[i, "Abundancia"]
    datos_binarios[localidad, especie] <- ifelse(abundancia > 0, 1, 0)
  }

  # Calcular la similitud de Jaccard entre las tres localidades
  similitud_jaccard <- matrix(NA, nrow = 3, ncol = 3)
  for (i in 1:2) {
    for (j in (i+1):3) {
      numerador <- sum(datos_binarios[i,] & datos_binarios[j,])
      denominador <- sum(datos_binarios[i,] | datos_binarios[j,])
      similitud_jaccard[i,j] <- numerador / denominador
      similitud_jaccard[j,i] <- similitud_jaccard[i,j]
    }
  }

  return(similitud_jaccard)
}

# Calcular la similitud de Jaccard entre las tres localidades específicas
similitud_jaccard_tres_localidades <- calcular_similitud_jaccard_tres_localidades(datos, "Casabito", "M

```

```

# Crear el mapa de calor utilizando heatmaply con paleta de colores RColorBrewer
heatmaply(similitud_jaccard_tres_localidades,
  symmetric = TRUE,
  labRow = c("Casabito", "Manaclar", "Arroyazo"),
  labCol = c("Casabito", "Manaclar", "Arroyazo"),
  main = "Mapa de Calor de Similitud de Jaccard entre Localidades",
  col = colorRampPalette(brewer.pal(n = 7, name = "YlOrRd"))(100), # Usar paleta YlOrRd de RCo
  fontsize_row = 12, # Tamaño de fuente para las etiquetas de fila
  fontsize_col = 12, # Tamaño de fuente para las etiquetas de columna
  showticklabels = TRUE, # Mostrar etiquetas en los ejes
  limits = c(0, 1), # Límites del colorbar
  na.rm = TRUE # Eliminar valores NA si los hay
)

```

Dominancia

```

# Cargar los datos desde el archivo RCEV.csv
ruta_archivo <- "C:/Users/jazmi/OneDrive/Documents/tesis-daniel/RCEV.csv"
datos <- read.csv(ruta_archivo)

# Definir función para calcular el Índice de Dominancia de Simpson
calcular_dominancia <- function(abundancia) {
  # Suma de cuadrados de los recuentos de especies
  suma_cuadrados <- sum(abundancia^2)

  # Total de individuos
  total_individuos <- sum(abundancia)

  # Índice de Dominancia de Simpson
  dominancia <- suma_cuadrados / (total_individuos^2)

  return(dominancia)
}

# Calcular la dominancia para cada localidad
dominancia_por_localidad <- aggregate(datos$Abundancia, by=list(Localidad=datos$Localidad), FUN=calcular_dominancia)

# Renombrar columnas
colnames(dominancia_por_localidad) <- c("Localidad", "Indice_Dominancia")

# Mostrar el resultado
print(dominancia_por_localidad)

library(ggplot2)

## Graficar la dominancia por localidad
ggplot(dominancia_por_localidad, aes(x=Localidad, y=Indice_Dominancia, fill=Indice_Dominancia)) +
  geom_bar(stat="identity") +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +
  labs(title="Dominancia de especies por localidad",
       x="Localidad", y="Índice de Dominancia") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  guides(fill = guide_legend(title = "Índice de Dominancia"))

```

```

ruta_archivo <- "C:/Users/jazmi/OneDrive/Documents/tesis-daniel/RCEV.csv"
datos <- read.csv(ruta_archivo)

library(heatmaply)

# Definir una función para calcular la distancia de Jaccard
calcular_distancia_jaccard <- function(matriz_binaria) {
  n <- nrow(matriz_binaria)
  distancia_jaccard <- matrix(NA, n, n)

  for (i in 1:(n - 1)) {
    for (j in (i + 1):n) {
      # Calcular el numerador (número de especies compartidas)
      numerador <- sum(matriz_binaria[i, ] & matriz_binaria[j, ])

      # Calcular el denominador (número de especies presentes en al menos una de las localidades)
      denominador <- sum(matriz_binaria[i, ] | matriz_binaria[j, ])

      # Calcular la distancia de Jaccard
      distancia_jaccard[i, j] <- 1 - (numerador / denominador)
      distancia_jaccard[j, i] <- distancia_jaccard[i, j]
    }
  }

  # Llenar la diagonal con ceros
  diag(distancia_jaccard) <- 0

  return(distancia_jaccard)
}

# Agregar prefijo único a los nombres duplicados
rownames(datos_binarios) <- make.unique(as.character(datos$Localidad))
colnames(datos_binarios) <- make.unique(as.character(datos$Localidad))

# Calcular la matriz de distancia de Jaccard
matriz_similitud_jaccard <- calcular_distancia_jaccard(datos_binarios)

# Graficar el mapa de calor de la matriz de similitud de Jaccard
heatmaply(matriz_similitud_jaccard,
  symmetric = TRUE, # Indica si la matriz es simétrica (en este caso, sí)
  labRow = rownames(datos_binarios), # Etiquetas de las filas
  labCol = colnames(datos_binarios), # Etiquetas de las columnas
  main = "Mapa de Calor de Similitud (Distancia de Jaccard)")

```