

Helechos a mansalva

Luis Daniel Fernández Montesinos

2025-01-09

Preliminares

- Cargar paquetes.

```
library(terra)
library(DT)
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)
library(ggrepel)
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)) %>%
  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') %>%
  group_by(Location) %>%
  mutate(`Transect number` = row_number()) %>%
  ungroup() %>%
  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', delete_dsn = T)

```

- Generar matriz de comunidad.

```

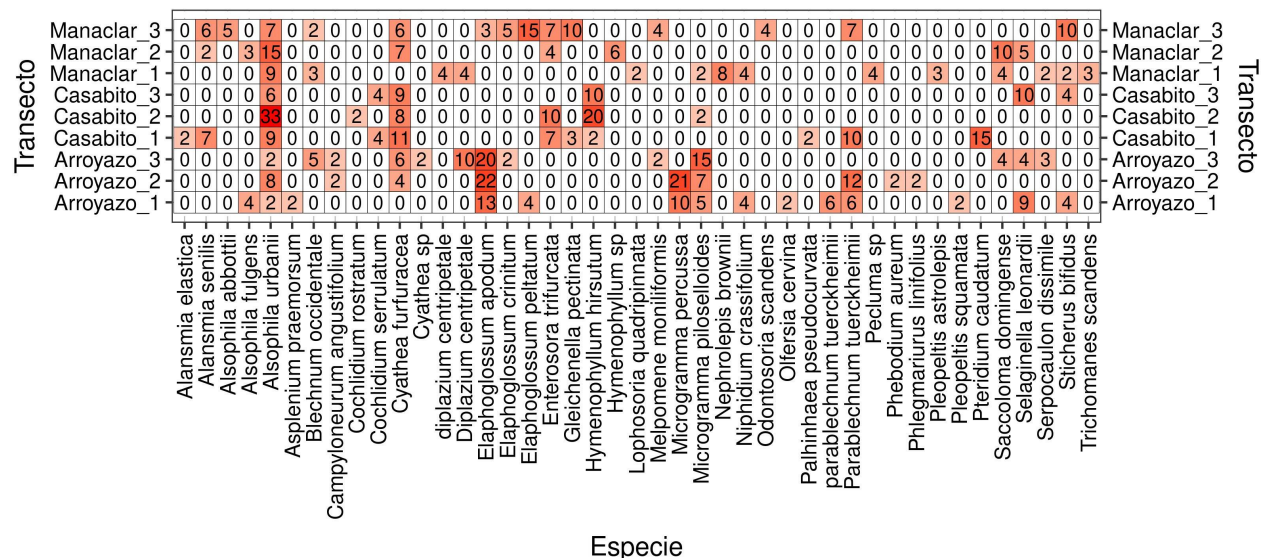
mc <- datos %>%
  pivot_wider(names_from = Especies, values_from = Abundancia, values_fill = 0) %>%
  column_to_rownames('id')

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.

##

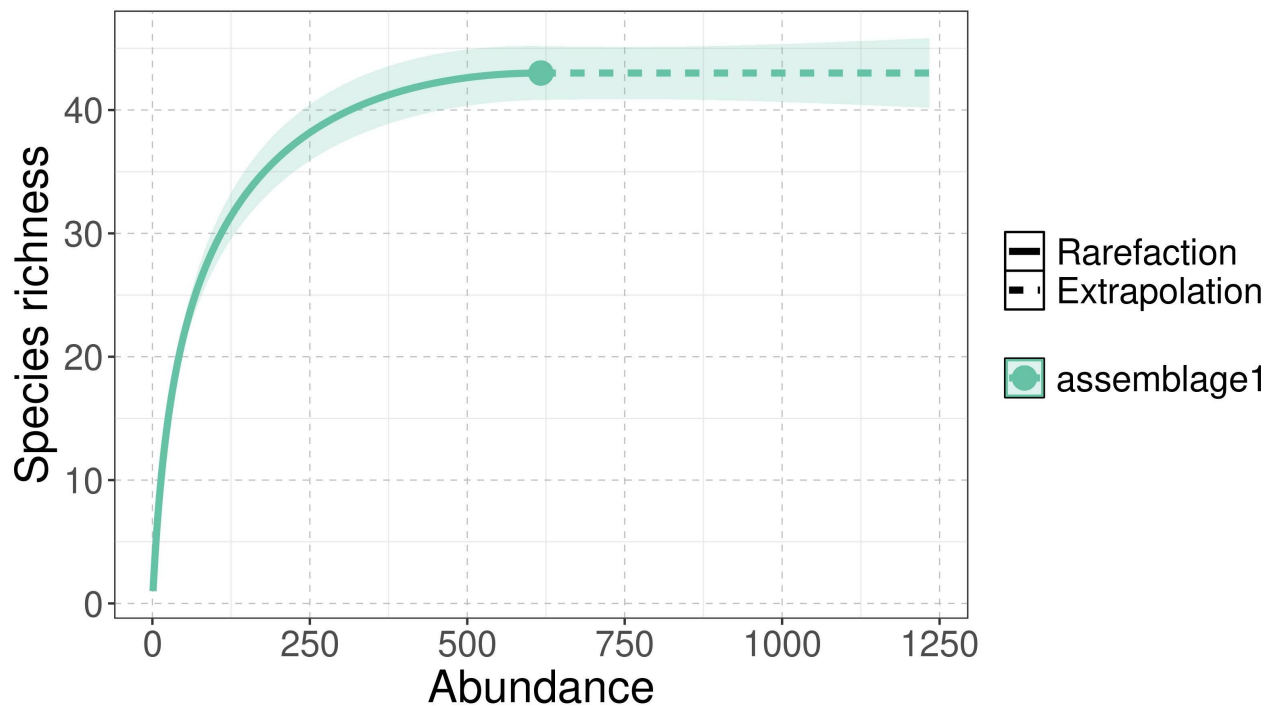
1st order jackknife: It uses the frequency of uniques to estimate the number of undetected species; see Chao (1987).

##

```
## 2nd order jackknife: It uses the frequencies of uniques and duplicates to estimate the number of und
##
## 95% Confidence interval: A log-transformation is used for all estimators so that the lower bound of
```

- 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
```

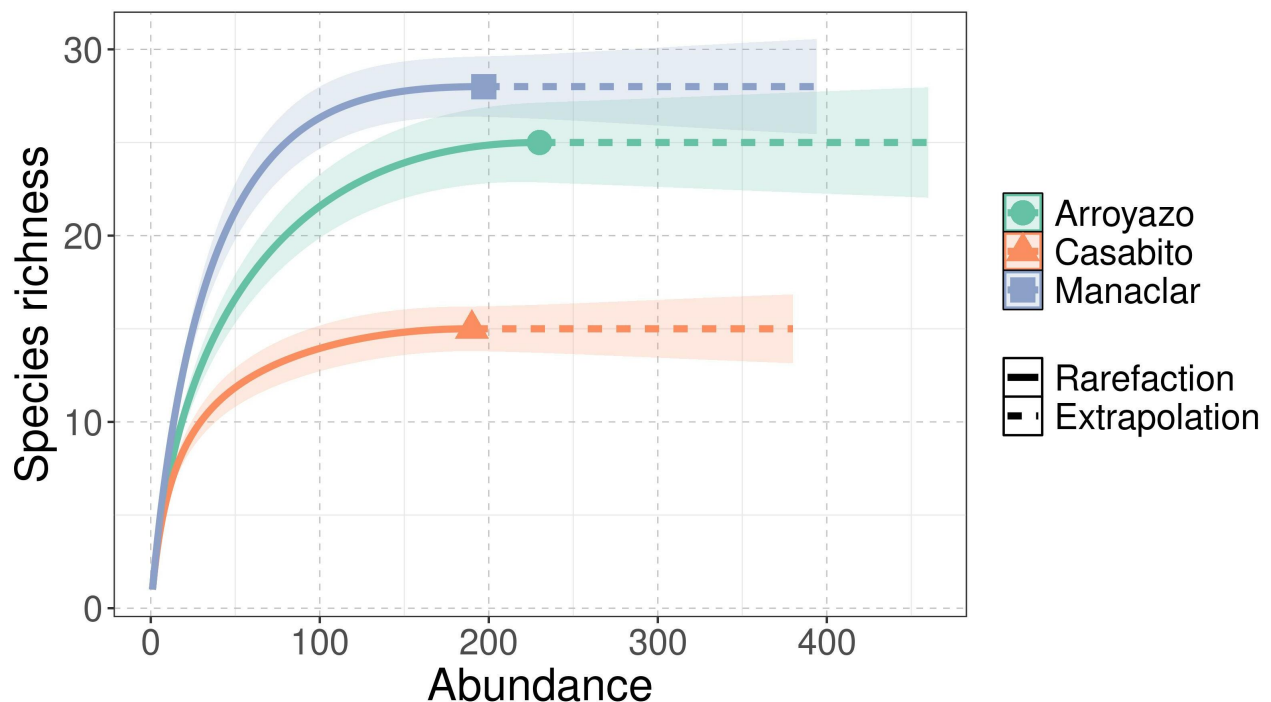


- Curva de acumulación por tipo de bosque

```

mc_grupos <- mc %>%
  mutate(g = grupos_seleccionados) %>%
  group_by(g) %>%
  summarise_all(sum) %>%
  column_to_rownames('g') %>%
  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

Explorar, aun siendo pocos transectos (en terreno tres transectos por formación es mucho, pero estadísticamente es poco), asociación con variables topográficas/geomorfológicas, cubierta y altura de dosel y elevación.

- Crear polígonos a partir de transectos

```
# Creación de segmentos de líneas
transect_lines_sf <- transect_coords_sf %>%
  group_by(id) %>%
  summarise(geometry = st_cast(st_combine(geometry), "LINESTRING"), .groups = "drop")

# Crear polígonos con buffer
transect_pol_sf_20m <- transect_lines_sf %>%
  st_transform(32619) %>%
  st_buffer(20) %>%
  st_transform(4326)
```

- Leer rásters

```
# Ruta a la carpeta que contiene los .tif
ruta_tifs <- "entradas"

# Leer todos los archivos tif (soporta multibanda automáticamente)
archivos_tif <- list.files(ruta_tifs, pattern = "\\\\.tif$", full.names = TRUE) %>%
  grep(pattern = paste('cobertura_esa|cobertura_copernicus_2019|geomorfonos',
    'formas_del_terreno|worldclim|heterogeneidad', 'MG_6073',
    sep = '|'),
    invert = T, value = T)

# EPSG deseado
crs_deseado <- "EPSG:4326"

# Leer y reprojectar si es necesario
raster_list <- lapply(archivos_tif, function(f) {
  r <- rast(f)
  if (crs(r) != crs_deseado) {
    r <- project(r, crs_deseado, method = 'near')
  }
  r
})

## |-----|-----|-----|-----|=====

# Leer los polígonos (asegúrate de que estén en EPSG:4326 también)
poligonos <- transect_pol_sf_20m |>
  vect() # convertir a objeto SpatVector de terra

# Categórico?
es_categorico <- c(FALSE, FALSE, FALSE, FALSE)

resultados <- Map(function(r, cat) {
  stat <- if (cat) "modal" else "mean"
  terra::extract(r, poligonos, fun = stat, na.rm = TRUE, bind = T)
}, raster_list, es_categorico)

# Añadir nombres basados en los archivos
names(resultados) <- basename(archivos_tif)

# Atributos
atributos_extraidos <- lapply(resultados, function(v) {
```

```

# Obtener solo los atributos adicionales (excluyendo el ID si es redundante)
as.data.frame(v)[, , drop = FALSE] # asume que la primera columna es ID o igual para todos
})

# Combinar
foo <- do.call(cbind, atributos_extraidos)
# Obtener el nombre de la primera columna que termina en 'id'
columna_id <- grep("id$", names(foo), value = TRUE)[1]
# Combinar final
atributos_combinados <- foo %>%
  rename(id = all_of(columna_id)) %>%
  rename_with(~ gsub("\\.tif\\.", '__', .x)) %>%
  rename_with(~ gsub('__discrete_classification', '', .x)) %>%
  rename_with(~ gsub('\\.', '_', .x)) %>%
  select(-ends_with('__id')) %>%
  rename_with(~ gsub('_b1.*', '', .x))

# Uniendo
env_sf <- poligonos %>%
  st_as_sf() %>%
  inner_join(atributos_combinados)

# Grupos
env <- env_sf %>%
  st_drop_geometry() %>%
  column_to_rownames('id')
env_grupos <- env %>%
  mutate(grupos = factor(gsub('_.*$', '', rownames(.)))) %>%
  pivot_longer(-c(grupos), names_to = "variable", values_to = "valor")

# Evaluando diferencias
env_grupos_ak <- env_grupos %>%
  group_by(variable) %>%
  summarise(
    p_valor_a = tryCatch(oneway.test(valor ~ grupos)$p.value, error = function(e) NA),
    p_valor_k = tryCatch(kruskal.test(valor ~ grupos)$p.value, error = function(e) NA)
  ) %>%
  arrange(p_valor_a)
# env_grupos_ak %>% estilo_kable(alinear = 'crr')
if(interactive()) {env_grupos_ak %>%
  datatable(options = list(pageLength = 140, autoWidth = FALSE),
    filter = 'top',) %>%
  formatSignif(columns = "p_valor_a", digits = 1) %>%
  formatSignif(columns = "p_valor_k", digits = 1)
}
env_grupos_ak %>% kableExtra::kable()

```

variable	p_valor_a	p_valor_k
srtm_v4_elevacion_elevation_elevation	0.0000010	0.0257064
geomorpho90m_topografia_stack_Compound_Topographic_Index	0.0030455	0.0509058
geomorpho90m_topografia_stack_Topographic_Position_Index	0.0041219	0.0664629
geomorpho90m_topografia_stack_Stream_Power_Index	0.0180972	0.0273237
geomorpho90m_topografia_stack_Terrain_Ruggedness_Index	0.0215116	0.0608101
geomorpho90m_topografia_stack_Vector_Ruggedness_Measure	0.0264424	0.0608101
wri_meta_canopy_height_cover_code	0.1355663	0.0608101
hansen_pct_cobertura_2000_treecover2000_treecover2000	0.1777834	0.1899457
geomorpho90m_topografia_stack_Roughness	0.5425836	0.4297961
geomorpho90m_topografia_stack_Slope	0.8376490	0.7326325

```
# Graficando
env_grupos_p <- env_grupos %>%
  ggplot() + aes(x = grupos, y = valor, group = grupos, fill = grupos) +
  geom_boxplot(lwd = 0.2) +
  scale_fill_brewer(palette = 'Set1') +
  theme_bw() +
  theme(legend.position="none") +
  facet_wrap(
    ~ variable,
    scales = 'free_y',
    ncol = 5,
    labeller = labeller(
      variable = function(x) label_wrap_gen(width = 35)(stringr::str_to_title(gsub("_", " ", x)))
    )
  )
# env_grupos_p
# Graficar
jpeg('salidas/env_grupos_p.jpg', width = 4000, height = 2000, res = 250)
env_grupos_p
dev.off()
```

pdf

2

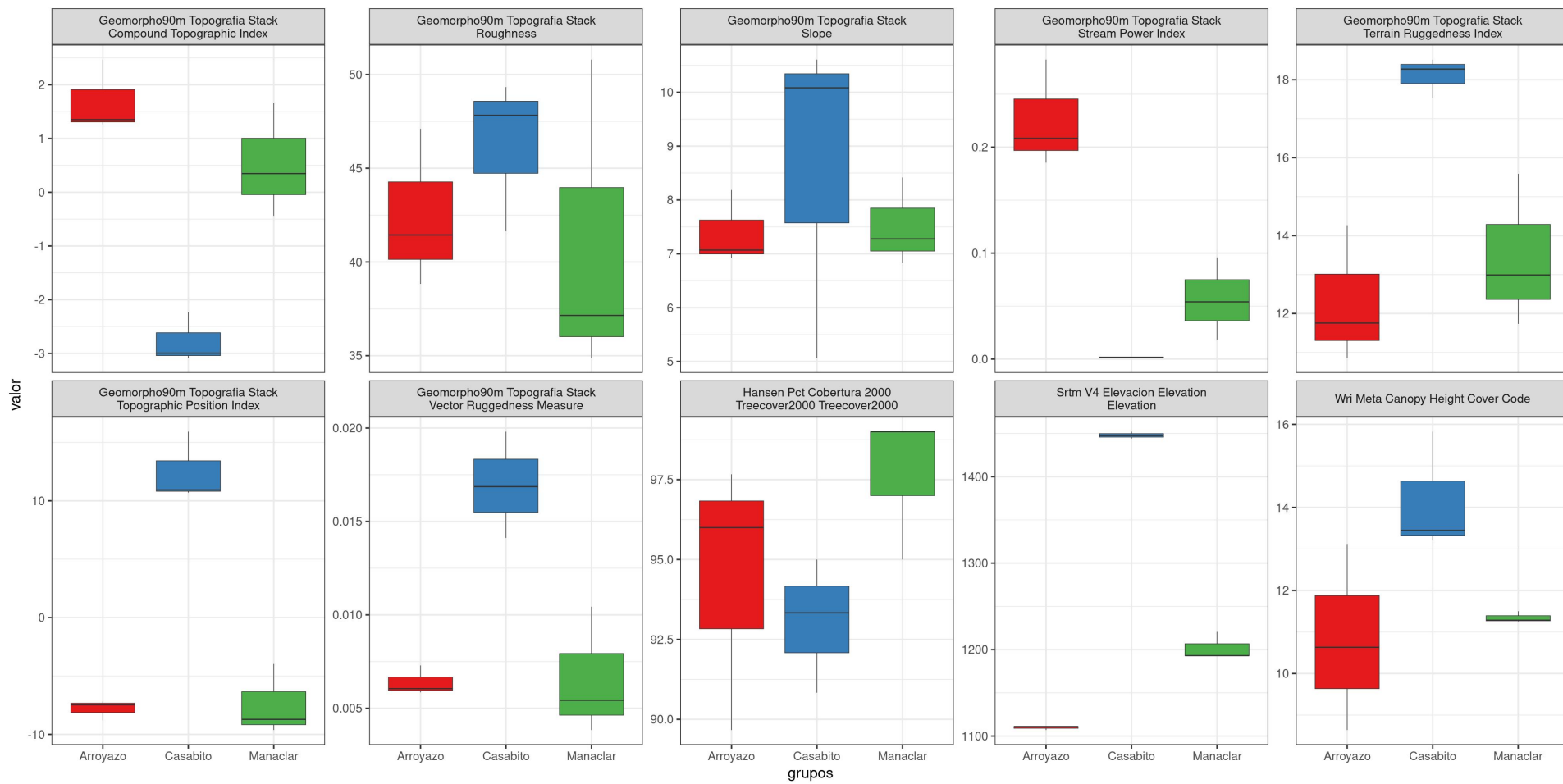


Figure 1: Variables ambientales por grupos

- Ordenación

```
mc_t <- decostand(mc, method = 'hellinger')
mc_t_ren <- mc_t %>%
  rename_all(~ paste('ESPECIE', .x))
env_spp <- env %>% bind_cols(mc_t_ren)
spp <- paste0(' ', grep('^ESPECIE', colnames(env_spp), value = T), ' ', collapse = ' + ')
my_formula <- as.formula(paste(spp, '~ .'))
set.seed(1); mod <- my_train(
  formula = my_formula,
  # preproceso = 'scale',
  data = env_spp,
  num_variables = 3:4)
print_my_train(mod)
```

\$resumen_variables

Subset selection object

10 Variables (and intercept)

	Forced in	Forced out
geomorpho90m_topografia_stack__Vector_Ruggedness_Measure	FALSE	FALSE
geomorpho90m_topografia_stack__Roughness	FALSE	FALSE
geomorpho90m_topografia_stack__Topographic_Position_Index	FALSE	FALSE
geomorpho90m_topografia_stack__Stream_Power_Index	FALSE	FALSE
geomorpho90m_topografia_stack__Compound_Topographic_Index	FALSE	FALSE
geomorpho90m_topografia_stack__Terrain_Ruggedness_Index	FALSE	FALSE
geomorpho90m_topografia_stack__Slope	FALSE	FALSE
hansen_pct_cobertura_2000_treecover2000__treecover2000	FALSE	FALSE
srtm_v4_elevacion_elevation_elevation	FALSE	FALSE
wri_meta_canopy_height__cover_code	FALSE	FALSE

1 subsets of each size up to 3

Selection Algorithm: 'sequential replacement'

```
geomorpho90m_topografia_stack__Vector_Ruggedness_Measure
1 ( 1 ) " "
2 ( 1 ) " "
3 ( 1 ) " "

geomorpho90m_topografia_stack__Roughness
1 ( 1 ) " "
2 ( 1 ) " "
3 ( 1 ) " "

geomorpho90m_topografia_stack__Topographic_Position_Index
1 ( 1 ) " "
2 ( 1 ) " "
3 ( 1 ) " "

geomorpho90m_topografia_stack__Stream_Power_Index
1 ( 1 ) " "
2 ( 1 ) " "
3 ( 1 ) "*"

geomorpho90m_topografia_stack__Compound_Topographic_Index
1 ( 1 ) " "
2 ( 1 ) " "
3 ( 1 ) " "

geomorpho90m_topografia_stack__Terrain_Ruggedness_Index
1 ( 1 ) " "
2 ( 1 ) " "
3 ( 1 ) " "
```

```

      geomorpho90m_topografia_stack__Slope
1 ( 1 ) " "
2 ( 1 ) " "
3 ( 1 ) " "
      hansen_pct_cobertura_2000_treecover2000__treecover2000
1 ( 1 ) " "
2 ( 1 ) "*"
3 ( 1 ) "*"
      srtm_v4_elevacion_elevation__elevation
1 ( 1 ) " "
2 ( 1 ) " "
3 ( 1 ) " "
      wri_meta_canopy_height__cover_code
1 ( 1 ) "*"
2 ( 1 ) "*"
3 ( 1 ) "*"

$resultados_nvmax
      nvmax      RMSE Rsquared      MAE  RMSESD RsquaredSD  MAESD
1      3 2.350682 0.3524792 2.200730 1.657974 0.5607735 1.799492
2      4 5.578701 0.5528418 4.731471 6.533586 0.3879218 5.919162

```

```
$mejor_ajuste
```

```

      nvmax
1      3

```

```

(covar <- grep(
  pattern = '\\(Intercept\\)',
  x = names(coef(mod$finalModel,unlist(mod$bestTune))),
  invert = T, value = T))

```

```

[1] "geomorpho90m_topografia_stack__Stream_Power_Index"
[2] "hansen_pct_cobertura_2000_treecover2000__treecover2000"
[3] "wri_meta_canopy_height__cover_code"

```

```

rda_mc_t <- rda(mc_t_ren %>% rename_all(~ gsub('^ESPECIE ', '', .)) ~ .,
  env %>% select_at(all_of(gsub('\\`', '', covar))), scale = T)

```

A continuación, el resumen del análisis de redundancia.

```
summary(rda_mc_t)
```

Call:

```
rda(formula = mc_t_ren %>% rename_all(~gsub("^ESPECIE ", "", .)) ~ geomorpho90m_topografia_stack__Slope, data = mc_t_ren, env = env, scale = T)
```

Partitioning of correlations:

	Inertia	Proportion
Total	43.00	1.0000
Constrained	16.19	0.3765
Unconstrained	26.81	0.6235

Eigenvalues, and their contribution to the correlations

Importance of components:

	RDA1	RDA2	RDA3	PC1	PC2	PC3	PC4

Eigenvalue	7.5991	4.9623	3.62895	9.1793	6.5401	4.5754	3.63655
Proportion Explained	0.1767	0.1154	0.08439	0.2135	0.1521	0.1064	0.08457
Cumulative Proportion	0.1767	0.2921	0.37652	0.5900	0.7421	0.8485	0.93307

PC5

Eigenvalue	2.87819
Proportion Explained	0.06693
Cumulative Proportion	1.00000

Accumulated constrained eigenvalues

Importance of components:

	RDA1	RDA2	RDA3
Eigenvalue	7.5991	4.9623	3.6290
Proportion Explained	0.4694	0.3065	0.2241
Cumulative Proportion	0.4694	0.7759	1.0000

Scaling 2 for species and site scores

- * Species are scaled proportional to eigenvalues
- * Sites are unscaled: weighted dispersion equal on all dimensions
- * General scaling constant of scores: 4.30665

Species scores

	RDA1	RDA2	RDA3	PC1	PC2
Elaphoglossum apodum	-0.575013	-0.14354	-0.082308	0.018937	-0.20179
Selaginella leonardii	-0.181456	0.05127	-0.090060	-0.100619	0.23302
Microgramma piloselloides	-0.434534	-0.21988	-0.244252	0.328637	-0.07620
Elaphoglossum peltatum	-0.153795	0.17165	0.276679	-0.226128	-0.35008
Asplenium praemorsum	-0.452894	0.13339	0.008416	-0.126701	0.22433
Niphidium crassifolium	-0.202220	0.38895	-0.007406	0.318980	0.18917
Pleopeltis squamata	-0.452894	0.13339	0.008416	-0.126701	0.22433
Microgramma percuta	-0.496516	-0.01982	0.157093	-0.066559	0.15815
Olfersia cervina	-0.452894	0.13339	0.008416	-0.126701	0.22433
Alsophila urbanii	0.456194	-0.23953	0.243548	0.109633	0.28199
Sticherus bifidus	0.031127	0.38541	0.143719	-0.057957	-0.13991
Parablechnum tuerckheimii	-0.240644	0.14810	0.073316	-0.403706	-0.08845
Alsophila fulgens	-0.329402	0.06721	0.235265	-0.091841	0.28936
Cyathea furfuracea	0.351204	-0.26139	-0.080914	-0.328289	-0.01583
Phebodium aureum	-0.247860	-0.11931	0.175854	0.014445	0.02110
Phlegmariurus linifolius	-0.247860	-0.11931	0.175854	0.014445	0.02110
Campyloneurum angustifolium	-0.341293	-0.26244	-0.179895	0.146810	-0.21806
Cyathea sp	-0.204073	-0.22686	-0.408265	0.178214	-0.30645
Saccoloma domingense	-0.007199	0.05878	0.067576	0.359479	0.02227
Melpomene moniliformis	-0.053609	-0.05427	-0.014446	-0.030823	-0.63122
Blechnum occidentale	-0.015698	0.13519	-0.193317	0.418137	-0.43640
Serpocaulon dissimile	-0.051855	0.09916	-0.324721	0.511673	-0.20423
Diplazium centripetale	-0.083780	0.04187	-0.357868	0.474713	-0.23460
Elaphoglossum crinitum	-0.041068	-0.04013	0.014070	-0.044642	-0.63273
parablechnum tuerckheimii	-0.452894	0.13339	0.008416	-0.126701	0.22433
Palhinhaea pseudocurvata	0.191778	0.15147	-0.305070	-0.366638	0.05909
Pteridium caudatum	0.191778	0.15147	-0.305070	-0.366638	0.05909
Alansmia elastica	0.191778	0.15147	-0.305070	-0.366638	0.05909
Enterosora trifurcata	0.377571	-0.21469	0.188771	-0.266721	-0.04528
Cochlidium serrulatum	0.295806	0.18190	-0.294397	-0.335697	0.17137

Alansmia senilis	0.219287	0.17151	0.101344	-0.403408	-0.20794
Gleichenella pectinata	0.191129	0.18382	0.098382	-0.360657	-0.42874
Hymenophyllum hirsutum	0.410554	-0.26186	-0.112239	-0.113519	0.25528
Cochlidium rostratum	0.266346	-0.47071	0.062675	0.082201	0.15551
Lophosoria quadripinnata	0.139822	0.36554	-0.016383	0.502845	0.04065
diplazium centripetale	0.139822	0.36554	-0.016383	0.502845	0.04065
Nephrolepis brownii	0.139822	0.36554	-0.016383	0.502845	0.04065
Pecluma sp	0.139822	0.36554	-0.016383	0.502845	0.04065
Pleopeltis astrolepis	0.139822	0.36554	-0.016383	0.502845	0.04065
Trichomanes scandens	0.139822	0.36554	-0.016383	0.502845	0.04065
Hymenophyllum sp	0.011112	-0.04221	0.299099	0.003516	0.15934
Alsophila abbottii	0.093470	0.11020	0.296751	-0.173448	-0.51096
Odontosoria scandens	0.093470	0.11020	0.296751	-0.173448	-0.51096

PC3

Elaphoglossum apodum	0.124266
Selaginella leonardii	-0.338692
Microgramma piloselloides	0.068380
Elaphoglossum peltatum	-0.286304
Asplenium praemorsum	-0.308463
Niphidium crassifolium	-0.177298
Pleopeltis squamata	-0.308463
Microgramma percussa	0.262834
Olfersia cervina	-0.308463
Alsophila urbanii	0.026700
Sticherus bifidus	-0.333285
Parablechnum tuerckheimii	0.308209
Alsophila fulgens	-0.260353
Cyathea furfuracea	0.084409
Phebodium aureum	0.527227
Phlegmariurus linifolius	0.527227
Campyloneurum angustifolium	0.344206
Cyathea sp	-0.066188
Saccoloma domingense	-0.039691
Melpomene moniliformis	-0.156532
Blechnum occidentale	-0.078163
Serpocaulon dissimile	-0.016053
Diplazium centripetale	-0.026514
Elaphoglossum crinitum	-0.157605
parablechnum tuerckheimii	-0.308463
Palhinhaea pseudocurvata	0.230699
Pteridium caudatum	0.230699
Alansmia elastica	0.230699
Enterosora trifurcata	-0.050049
Cochlidium serrulatum	0.046366
Alansmia senilis	0.075811
Gleichenella pectinata	0.007054
Hymenophyllum hirsutum	-0.137601
Cochlidium rostratum	-0.132602
Lophosoria quadripinnata	0.046375
diplazium centripetale	0.046375
Nephrolepis brownii	0.046375
Pecluma sp	0.046375
Pleopeltis astrolepis	0.046375
Trichomanes scandens	0.046375

Hymenophyllum sp	-0.039454
Alsophila abbottii	-0.134248
Odontosoria scandens	-0.134248

Site scores (weighted sums of species scores)

	RDA1	RDA2	RDA3	PC1	PC2	PC3
Arroyazo_1	-3.0355	1.2147	0.7450	-0.78332	1.3869	-1.9070
Arroyazo_2	-1.5277	-1.3446	0.5624	0.08930	0.1304	3.2595
Arroyazo_3	-1.1528	-1.1983	-2.9200	1.10179	-1.8946	-0.4092
Casabito_1	1.6067	0.3205	-1.4231	-2.26671	0.3653	1.4263
Casabito_2	1.3724	-2.1195	0.5447	0.50820	0.9614	-0.8198
Casabito_3	0.9347	-0.3609	-0.4221	-0.70748	0.9731	-0.7626
Manaclar_1	0.6477	3.5339	-1.0170	3.10880	0.2513	0.2867
Manaclar_2	0.6055	-0.7106	1.5912	0.02174	0.9851	-0.2439
Manaclar_3	0.5490	0.6648	2.3389	-1.07233	-3.1589	-0.8300

Site constraints (linear combinations of constraining variables)

	RDA1	RDA2	RDA3	PC1	PC2	PC3
Arroyazo_1	-2.8000	0.8247	0.05203	-0.78332	1.3869	-1.9070
Arroyazo_2	-1.5324	-0.7376	1.08720	0.08930	0.1304	3.2595
Arroyazo_3	-1.2617	-1.4025	-2.52406	1.10179	-1.8946	-0.4092
Casabito_1	1.1856	0.9364	-1.88607	-2.26671	0.3653	1.4263
Casabito_2	1.6467	-2.9101	0.38748	0.50820	0.9614	-0.8198
Casabito_3	1.2507	0.6089	-0.69908	-0.70748	0.9731	-0.7626
Manaclar_1	0.8644	2.2599	-0.10129	3.10880	0.2513	0.2867
Manaclar_2	0.0687	-0.2610	1.84916	0.02174	0.9851	-0.2439
Manaclar_3	0.5779	0.6813	1.83463	-1.07233	-3.1589	-0.8300

Biplot scores for constraining variables

	RDA1	RDA2	RDA3
geomorpho90m_topografia_stack__Stream_Power_Index	-0.9943	-0.1008	0.03516
hansen_pct_cobertura_2000_treecover2000__treecover2000	-0.1133	0.0765	0.99061
wri_meta_canopy_height__cover_code	0.7396	-0.5754	-0.34913

	PC1	PC2	PC3
geomorpho90m_topografia_stack__Stream_Power_Index	0	0	0
hansen_pct_cobertura_2000_treecover2000__treecover2000	0	0	0
wri_meta_canopy_height__cover_code	0	0	0

El factor de inflación de la varianza.

```
vif.cca(rda_mc_t)
```

```

      geomorpho90m_topografia_stack__Stream_Power_Index
                        2.085908
hansen_pct_cobertura_2000_treecover2000__treecover2000
                        1.410378
      wri_meta_canopy_height__cover_code
                        2.636798

```

Represento el gráfico *triplot*.

```

# Triplot
escalado <- 1
plot(rda_mc_t,
     scaling = escalado,
     display = c("lc", "cn"),
     main = paste("Triplot de RDA especies ~ variables, escalamiento", escalado)
)
rda_mc_t_scl <- scores(rda_mc_t,
                      choices = 1:2,
                      scaling = escalado,
                      display = "sp"
)
text(rda_mc_t, "species", col="red", cex=0.8, scaling=escalado)
arrows(0, 0,
       rda_mc_t_scl[, 1] * 0.9,
       rda_mc_t_scl[, 2] * 0.9,
       length = 0,
       lty = 1,
       col = "red"
)

```

[illegible]

- ```
Extraer coordenadas
escalado <- 1
scores_rda <- scores(rda_mc_t, scaling = escalado)

Sitios (lc: linear combinations)
sitios <- as_tibble(scores_rda$sites, rownames = "Sitio")

Variables explicativas (cn: canonical axes)
variables <- as_tibble(scores_rda$biplot, rownames = "Variable")

Estimar máximo de los sitios
max_sitios <- max(abs(c(sitios$RDA1, sitios$RDA2)))
```

```

Estimar máximo de vectores
max_vectores <- max(abs(c(variables$RDA1, variables$RDA2)))

Escalar vectores para que quepan proporcionalmente
escala_vectores <- max_sitios / max_vectores * 0.9

Renombrar
variables <- variables %>%
 mutate(across(c(RDA1, RDA2), ~ . * escala_vectores)) %>%
 mutate(Variable = case_when(
 Variable == "geomorpho90m_topografia_stack_Stream_Power_Index" ~ "Stream Power Index",
 Variable == "hansen_pct_cobertura_2000_treecover2000__treecover2000" ~ "Tree Cover 2000",
 Variable == "wri_meta_canopy_height__cover_code" ~ "Canopy Height",
 TRUE ~ Variable
))

Especies (sp: species scores)
especies <- as_tibble(scores(rda_mc_t, display = "species", scaling = escalado), rownames = "Especie")

Gráfico base
triplot_ggplot <- ggplot() +
 # Sitios
 geom_point(data = sitios, aes(x = RDA1, y = RDA2), color = "black") +
 geom_text(data = sitios, aes(x = RDA1, y = RDA2, label = Sitio), hjust = 0.5, vjust = -0.5, size = 3)

 # Especies
 # geom_text(data = especies,
 # aes(x = RDA1, y = RDA2, label = Especie),
 # color = "red", size = 3) +
 geom_text_repel(data = especies,
 aes(x = RDA1, y = RDA2, label = Especie),
 color = "red", size = 3, max.overlaps = 50) +

 # Variables ambientales
 geom_segment(data = variables,
 aes(x = 0, y = 0, xend = RDA1, yend = RDA2),
 arrow = arrow(length = unit(0.2, "cm")),
 color = "blue", linewidth = 0.6) +
 geom_text(data = variables,
 aes(x = RDA1, y = RDA2, label = Variable),
 color = "blue", hjust = 0.5, vjust = -0.5, size = 3) +

 # Ejes
 geom_hline(yintercept = 0, linetype = "dotted") +
 geom_vline(xintercept = 0, linetype = "dotted") +
 labs(title = paste("RDA, triplot species ~ variables, scaling", escalado),
 x = "RDA1", y = "RDA2") +
 coord_equal() +
 theme_minimal()

Graficar
jpeg('salidas/triplot_ggplot.jpg', width = 3000, height = 3000, res = 350)
triplot_ggplot

```

```
dev.off()
```

```
pdf
```

```
2
```

RDA, triplot species ~ variables, scaling 1



Figure 2: Triplot en ggplot

## 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)")

```