

# Análisis Multivariados en R

Curso Colaborativo IIAP - UNAMAD

Irwing S. Saldaña

Instituto de Ciencias Antonio Brack

Departamento de Ecoinformática y Biogeografía

Perú, 2021



# Blgo. Irwing S. Saldaña

## Instructor

Dpto. de Ecoinformática y Biogeografía,  
Instituto de Ciencias Antonio Brack, Perú

[Website](#) | [ResearchGate](#) | [Linkedin](#) | [R Latam Blog](#) | [Github](#)



# Análisis Canónico

[ Análisis Multivariados con restricciones o constreñidos ]



# Análisis Canónico



# Recordemos el Análisis de Correspondencia

Implica métodos en los que constreñimos o condicionamos una matriz de datos Y (normalmente de biodiversidad) con una matriz de datos X (normalmente de ambientales). Recordemos el análisis de correspondencia (CA):

```
especies <- read.csv("bases/cca_spe.csv", row.names=1, sep=";")  
especies
```

Arctlute	Pardlugu	Zoraspin	Pardnigr	Pardpull	AuIalbi	Trocterr	Alopocene	Pardmont	Alopacce	Alopfabr	Arctperi
0	2	1	0	0	0	5	0	0	0	0	0
0	3	1	1	0	0	4	1	0	0	0	0
0	3	1	0	0	0	4	1	0	0	0	0
0	2	2	1	0	0	5	1	0	0	0	0
0	1	1	0	0	0	4	0	0	0	0	0
0	2	0	0	0	0	5	1	0	0	0	0



# Recordemos el Análisis de Correspondencia

Implica métodos en los que constreñimos o condicionamos una matriz de datos Y (normalmente de biodiversidad) con una matriz de datos X (normalmente de ambientales). Recordemos el análisis de correspondencia (CA):

```
library(ca)
especies_log <- log1p(especies)
CA <- ca(especies_log)
summary(CA)
```

```
##
## Principal inertias (eigenvalues):
##
##   dim    value      %   cum%   scree plot
##   1     0.558173  53.7  53.7  *****
##   2     0.231494  22.3  76.0  ****
##   3     0.117723  11.3  87.4  ***
##   4     0.038470   3.7  91.1  *
##   5     0.027822   2.7  93.7  *
##   6     0.020520   2.0  95.7
```



# Análisis de Correspondencia Canónica (CCA)

[ Incorporando una matriz X en el análisis de CA ]



# Análisis de Correspondencia Canónica (CCA)

El CCA es la versión constreñida del CA.

Se crea utilizando la función `cca()`

Estructura del código en R:

```
# Modo solo matrices  
CCA <- cca(matriz_X, matriz_Y)  
  
# Modo fórmula (mucho más versátil)  
CCA <- cca(matriz_Y ~ ., data= matriz_X)
```

- Utiliza distancias de **Chi cuadrado**.
- Recuerda que las *especies raras se ven sobreexpresadas* (**logaritmizar** para reducir el peso de ellas).
- Sirve para **identificar la proporción de varianza** de una matriz de biodiversidad (matriz de conteos) que es **explicada** por una matriz de datos ambientales (matriz de mediciones).



# Análisis de Correspondencia Canónica (CCA)

```
ambiente <- read.csv("bases/cca_env.csv", row.names=1, sep=";")  
CCA <- cca(especies_log ~ ., data = ambiente)  
summary(CCA)
```

```
##  
## Call:  
## cca(formula = especies_log ~ WaterContent + BareSand + CoverMoss +      LightReflection + FallenT  
##  
## Partitioning of scaled Chi-square:  
##           Inertia Proportion  
## Total      1.0386   1.0000  
## Constrained 0.7695   0.7409  
## Unconstrained 0.2691   0.2591  
##  
## Eigenvalues, and their contribution to the scaled Chi-square  
##  
## Importance of components:  
##           CCA1    CCA2    CCA3    CCA4    CCA5    CCA6    CA1  
## Eigenvalue  0.510 0.1982 0.04310 0.01244 0.004102 0.001689 0.1110
```



# Test de Permutaciones: modelo

- Necesario para validar si la varianza explicada por la matriz ambiental explica más de lo que podría explicar por azar.

```
# Probando la significancia del modelo CCA:  
anova.cca(cca)
```

```
## Permutation test for cca under reduced model  
## Permutation: free  
## Number of permutations: 999  
##  
## Model: cca(formula = especies_log ~ WaterContent + BareSand + CoverMoss + LightReflection + FallenLeaves + SpeciesRichness)  
##          Df ChiSquare      F Pr(>F)  
## Model      6   0.76951 10.008  0.001 ***  
## Residual  21   0.26911  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



# Test de Permutaciones: términos

- Necesario para validar si la varianza explicada por la matriz ambiental explica más de lo que podría explicar por azar.

```
# Probando la significancia de los término (variables ambientales):  
anova.cca(cca, by="terms")
```

```
## Permutation test for cca under reduced model  
## Terms added sequentially (first to last)  
## Permutation: free  
## Number of permutations: 999  
##  
## Model: cca(formula = species_log ~ WaterContent + BareSand + CoverMoss + LightReflection + Falle  
##                 Df ChiSquare      F Pr(>F)  
## WaterContent     1   0.45422 35.4459  0.001 ***  
## BareSand         1   0.06130  4.7839  0.007 **  
## CoverMoss        1   0.07015  5.4744  0.003 **  
## LightReflection  1   0.08363  6.5261  0.002 **
```



# Test de Permutaciones: ejes canónicos

- Necesario para validar si la varianza explicada por la matriz ambiental explica más de lo que podría explicar por azar.

```
# Probando la significancia de los ejes CCA (CCA1 y CCA2 al menos deben ser significativos):  
anova.cca(CCA, by="axis")
```

```
## Permutation test for cca under reduced model  
## Forward tests for axes  
## Permutation: free  
## Number of permutations: 999  
##  
## Model: cca(formula = especies_log ~ WaterContent + BareSand + CoverMoss + LightReflection + Falle  
##              Df ChiSquare      F Pr(>F)  
## CCA1       1   0.50998 39.7973  0.001 ***  
## CCA2       1   0.19820 15.4667  0.001 ***  
## CCA3       1   0.04310  3.3630  0.168  
## CCA4       1   0.01244  0.9708  0.908
```



# Gráfico Final CCA

```
library(ggvegan)  
autoplots(CCA)
```



# Análisis de Redundancia (RDA)

[ La extensión canónica del PCA ]



# Análisis de Redundancia (RDA)

El RDA es la versión constreñida del PCA.

Se crea utilizando la función `rda()`

Estructura del código en R:

```
# Modo solo matrices  
RCA <- rca(matriz_X, matriz_Y)  
  
# Modo fórmula (mucho más versátil)  
RCA <- rca(matriz_Y ~ ., data= matriz_X)
```

- Utiliza distancias **Euclídeas**. Pero se puede utilizar la misma propiedad de las distancias de Hellinger o distancias Chord que en el PCA.
- Igual que el CCA sirve para **identificar la proporción de varianza** de una matriz de dependiente (matriz de conteos o "biodiversidad" pero transformada, o una matriz de mediciones) que es **explicada** por una matriz de datos independientes (matriz de mediciones).



# Tipos de Análisis de Redundancia (RDA)

- **RDA clásico (con distancias euclídeas)**

- Usa PCA como motor de análisis multivariado
  - Trabaja con regresiones múltiples también

- **RDA basado en transformaciones (tb-RDA)**

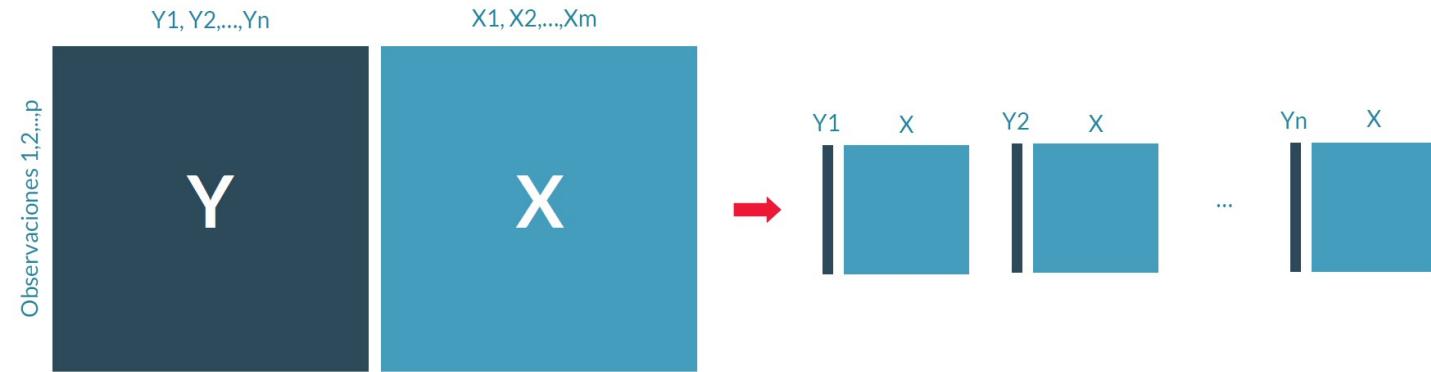
- Usa PCA como motor de análisis multivariado
  - Trabaja con regresiones múltiples también

- **RDA basado en distancias (db-RDA)**

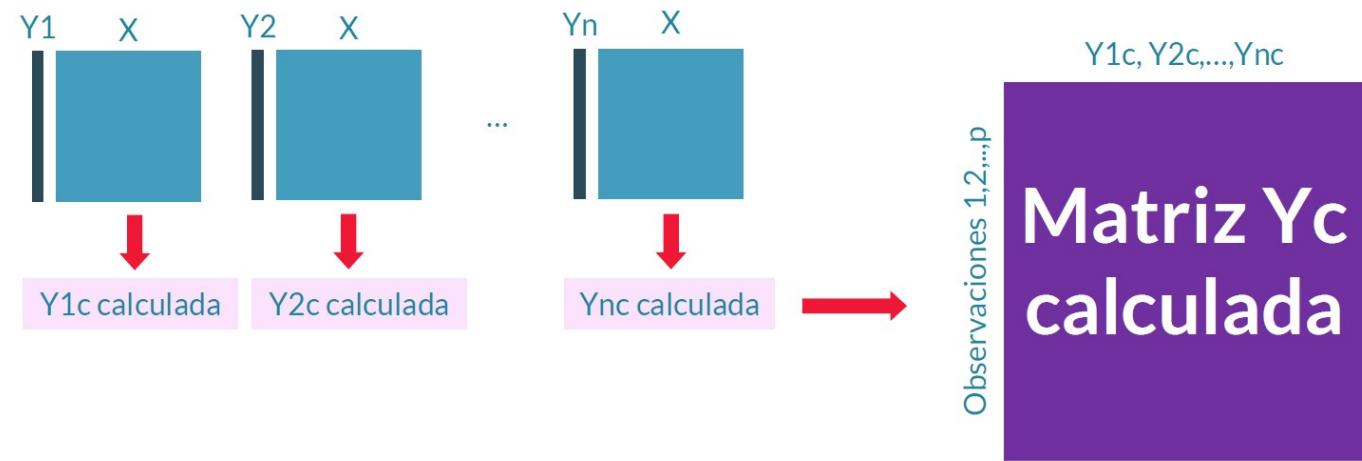
- Usa PCoA (MDS) como motor de análisis multivariado
  - Trabaja con regresiones múltiples también
  - Se requiere evitar eigenvalores negativos (aplicar raíz cuadrada a las distancias Bray Curtis para este fin)

...Veamos la estrategia de proyección del RDA



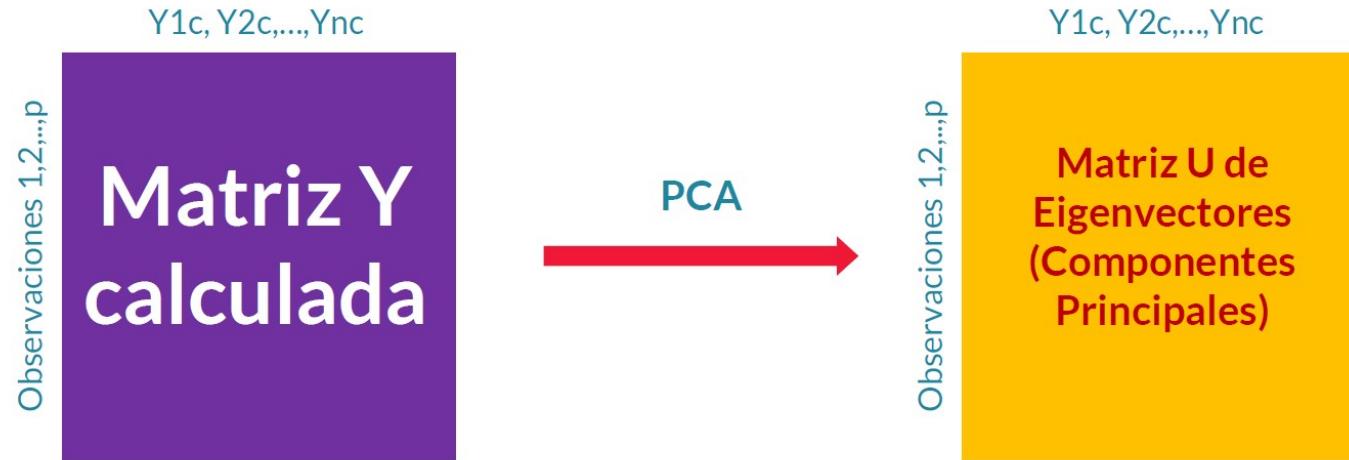


> Blgo. Irwing S Saldaña [Programa de Certificación Especializado Data Science: Estadística y Análisis de Datos en R]

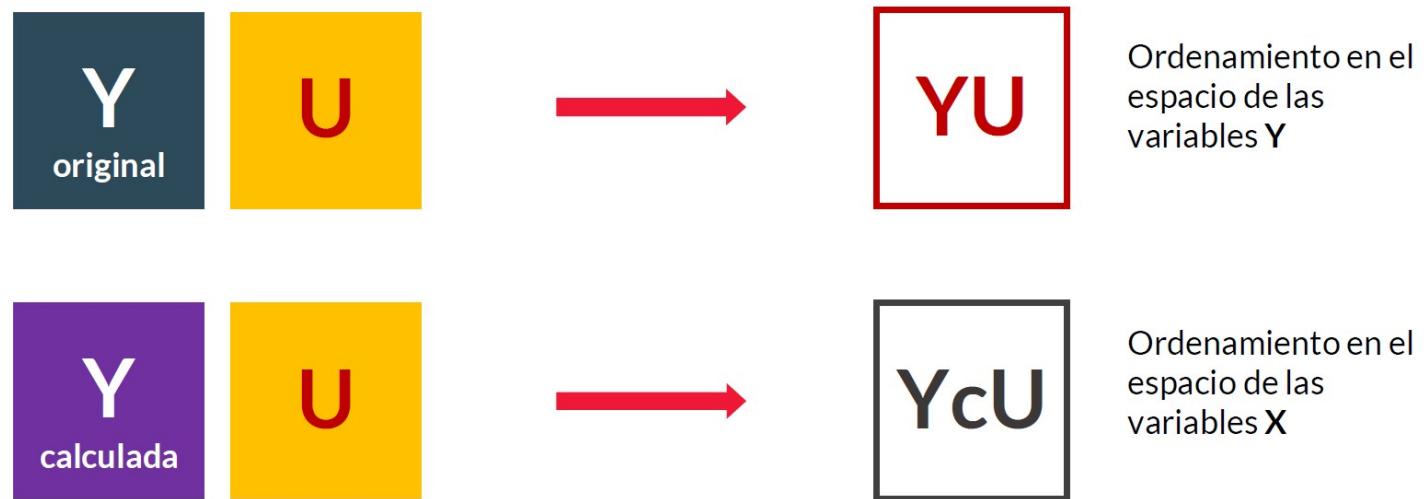


> Blgo. Irwing S Saldaña [Programa de Certificación Especializado Data Science: Estadística y Análisis de Datos en R]



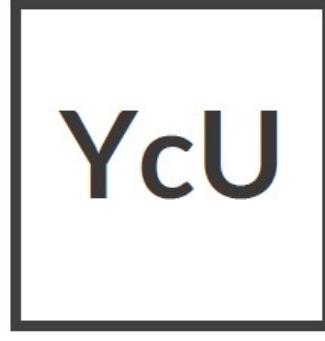


> Blgo. Irwing S Saldaña [Programa de Certificación Especializado Data Science: Estadística y Análisis de Datos en R]





**YU**



**YcU**



**Triplot**

Ordenamiento  
en el espacio de  
las variables Y

Ordenamiento  
en el espacio de  
las variables X



# Recomendaciones

- Piensa en el RDA como si fueras a realizar un PCA.
- Cumple con los requerimientos del PCA, transforma las matrices si se requiere, o estandariza si es necesario.

```
# Si se requiere escalar o no, se activa o desactiva este argumento  
rda(..., scale=TRUE)
```

- Si la matriz dependiente es de abundancias: aplica tb-RDA o db-RDA

```
# RDA basado en transformaciones  
rda(..., scale=FALSE)
```

```
# RDA basado en distancias  
dbrda(..., distance = "bray")
```



# Recomendaciones

- Recuerda que no toda la varianza de la matriz Y es explicada por la matriz X, así que busquemos valores altos de varianza constreñida.
- Caso contrario, no es correcto hacer una análisis canónico, sino un no restringido, o con regresiones lineales (o generalizadas) múltiples.
- No pueden haber igual o más variables explicativas que observaciones.



# Escalas

- `scaling = 1`  $\implies$  Se centra en sitios (filas), escala los scores de los sitios por  $\lambda_i$
- `scaling = 2`  $\implies$  Se centra en especies (columnas Y), escala los scores de las especies por  $\lambda_i$
- `scaling = 3`  $\implies$  Escalado simétrico, escala ambos scores por  $\sqrt{\lambda_i}$
- `scaling = 0`  $\implies$  scores crudos

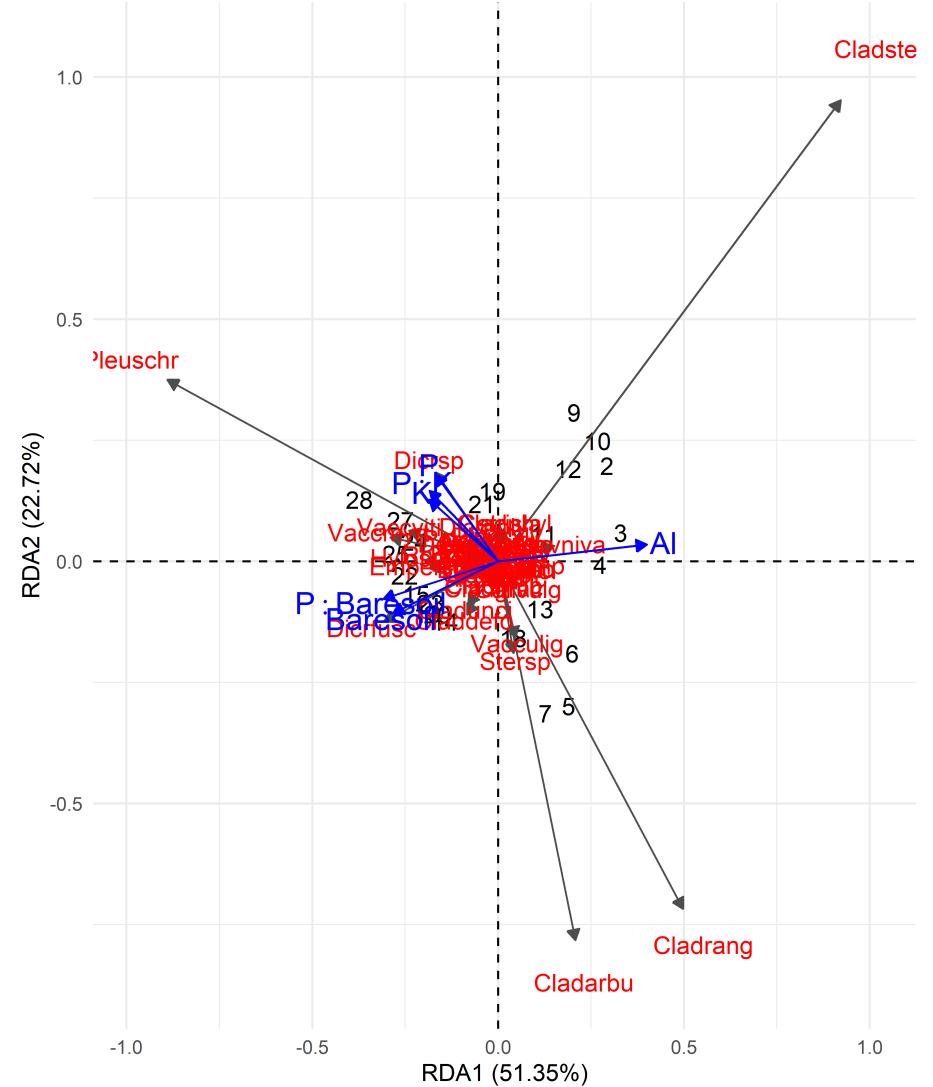
```
# Observar scores individuales
scores(RDA, choices = 1:2, display = "species", scaling = 3)
```

- `scaling = 1`  $\implies$  "sites"
- `scaling = 2`  $\implies$  "species"
- `scaling = 3`  $\implies$  "symmetric"
- `scaling = 0`  $\implies$  "none"



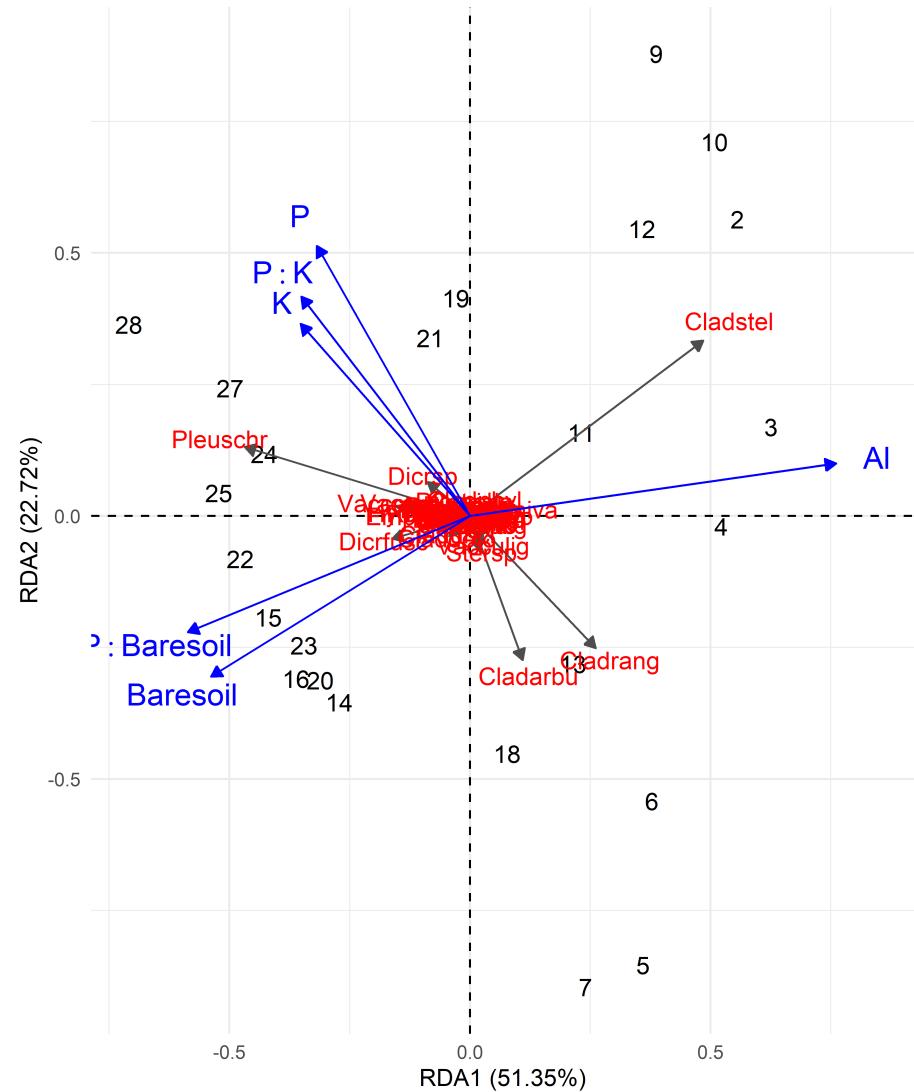
# Sites scaling = 1

- Las distancias entre las observaciones son aproximadamente euclídeas. Objetos cercanos se asumen contienen variables con similar valor.
- La distancia en ángulos rectos entre las observaciones y las variables (X o Y) reflejan que están asociados a ellos.
- Los ángulos entre las variables de la matriz Y no son interpretables.
- El coseno de los ángulos entre los vectores de las variables de la matriz Y y X reflejan su nivel de correlación (e.g.  $\cos(90\pi/180) = 0$ ,  $\cos(20\pi/180) = 0.94$ ).

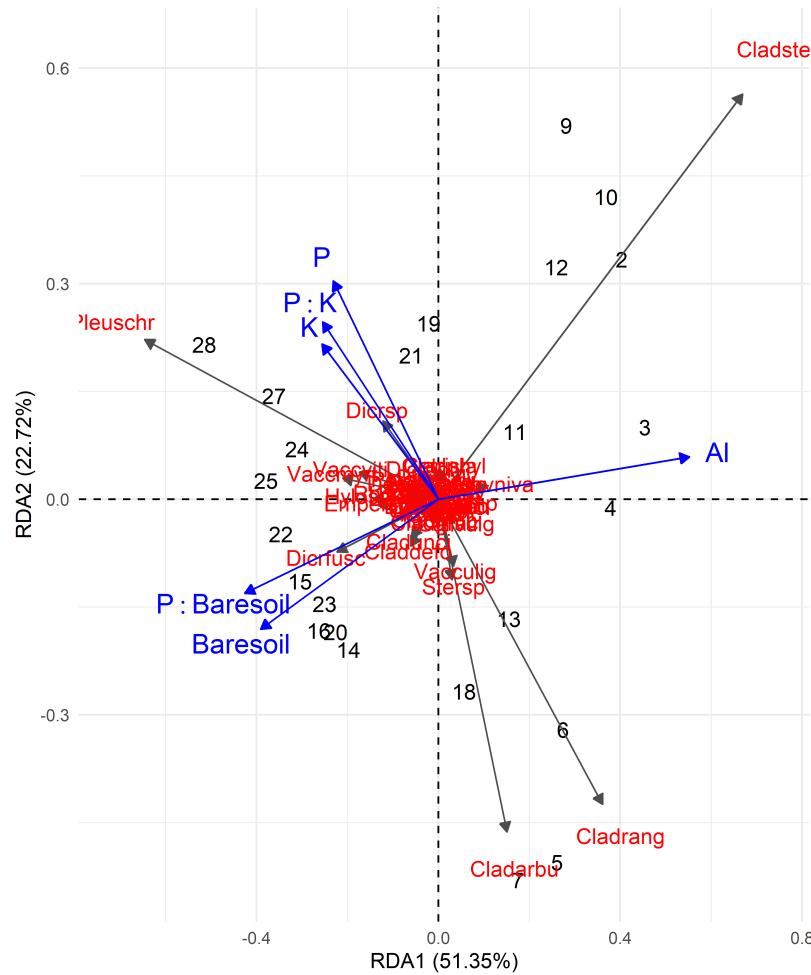


# Species scaling = 2

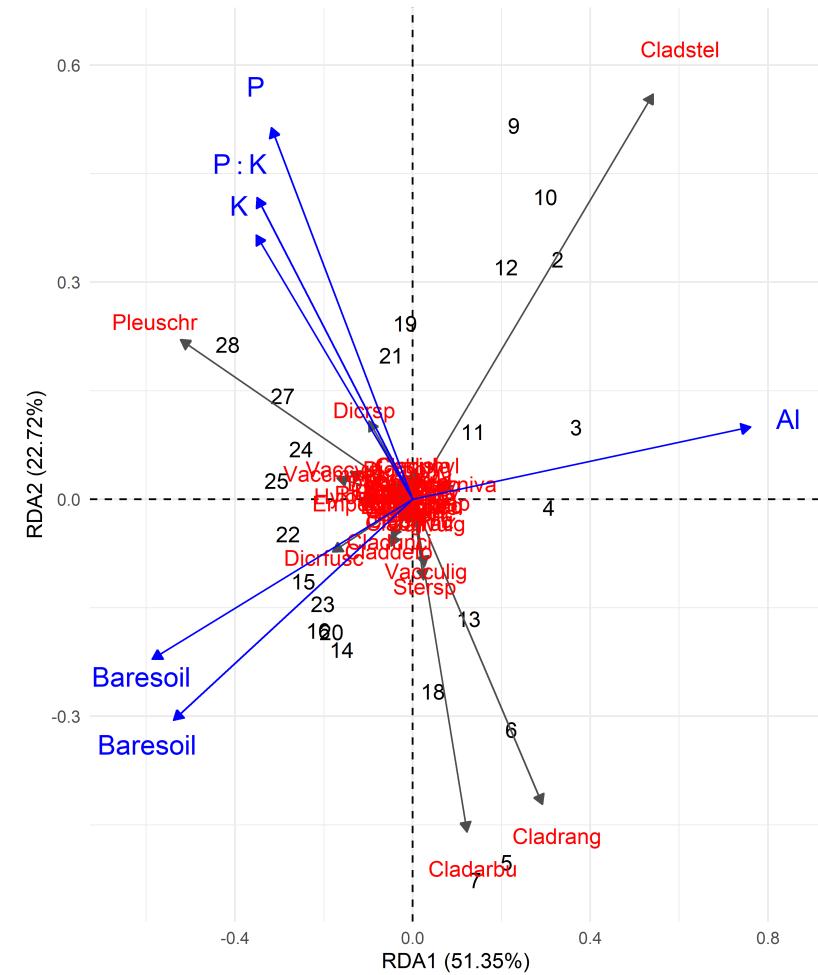
- Las distancias entre las observaciones no son euclídeas.
- La distancia en ángulos rectos entre las observaciones y las variables (X o Y) reflejan que están asociados a ellos.
- El coseno de los ángulos entre todos los vectores (X o Y) reflejan su nivel de correlación (e.g.  $\cos(90\pi/180) = 0$ ,  $\cos(20\pi/180) = 0.94$ ).



Species scaling = 3



Species scaling = 0



# Modelamiento de RDA

[ También aplicado al CCA ]



# Modelamiento con RDA - CCA

Dado que estos métodos utilizan regresiones lineales en segundo plano para poder hacer el ajuste del modelo, las mismas recomendaciones que para un LM se deben aplicar aquí.

- Identificar las variables importantes y solo incluirlas a ellas.
- Si se conoce que la relación entre una variable y las especies tiene **comportamiento cuadrático o cúbico**, se puede incluir ello en la fórmula del modelo RDA.
- Si se sabe que **dos variables interactúan** y que dicha interacción influencia a la presencia de especies, también se puede incluir ello en la fórmula del modelo RDA.
- Procedimientos de **selección automática de variables** deben ser consideradas con cuidado.



```
library(vegan)
data("varespec")
data("varechem")
RDA_vare <- rda(decostand(varespec, "hellinger") ~ Al + K + Baresoil, data = varechem)
RDA_vare

## Call: rda(formula = decostand(varespec, "hellinger") ~ Al + K +
## Baresoil, data = varechem)
##
##          Inertia Proportion Rank
## Total      0.3647    1.0000
## Constrained 0.1319    0.3616    3
## Unconstrained 0.2328    0.6384   20
## Inertia is variance
##
## Eigenvalues for constrained axes:
##      RDA1     RDA2     RDA3
## 0.10019 0.02541 0.00625
##
## Eigenvalues for unconstrained axes:
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8
## 0.08502 0.03623 0.02715 0.02268 0.01208 0.01067 0.00998 0.00701
## (Showing 8 of 20 unconstrained eigenvalues)
```



```
library(vegan)
data("varespec")
data("varechem")
RDA_vare <- rda(decostand(varespec, "hellinger") ~ Al + P + K + Baresoil, data = varechem)
RDA_vare
```

```
## Call: rda(formula = decostand(varespec, "hellinger") ~ Al + P + K +
## Baresoil, data = varechem)
##
##          Inertia Proportion Rank
## Total      0.3647    1.0000
## Constrained 0.1494    0.4098    4
## Unconstrained 0.2152    0.5902   19
## Inertia is variance
##
## Eigenvalues for constrained axes:
##      RDA1     RDA2     RDA3     RDA4
## 0.10067 0.02863 0.01501 0.00513
##
## Eigenvalues for unconstrained axes:
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8
## 0.08319 0.03445 0.02427 0.01639 0.01160 0.01047 0.00943 0.00639
## (Showing 8 of 19 unconstrained eigenvalues)
```



```

library(vegan)
data("varespec")
data("varechem")
RDA_vare <- rda(decostand(varespec, "hellinger") ~ Al + P*(K + Baresoil), data = varechem)
RDA_vare

## Call: rda(formula = decostand(varespec, "hellinger") ~ Al + P * (K +
## Baresoil), data = varechem)
##
##          Inertia Proportion Rank
## Total      0.3647    1.0000
## Constrained 0.1968    0.5395    6
## Unconstrained 0.1679    0.4605   17
## Inertia is variance
##
## Eigenvalues for constrained axes:
##      RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## 0.10104 0.04470 0.01707 0.01509 0.01381 0.00505
##
## Eigenvalues for unconstrained axes:
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8
## 0.06484 0.02672 0.01664 0.01514 0.01038 0.00852 0.00660 0.00458
## (Showing 8 of 17 unconstrained eigenvalues)

```



# Enfoques automáticos de selección de modelos

- Define el mejor modelo con todas las posibles variables importantes (Modelo Full)
- Utiliza el método de selección corregido para RDA-CCA (usando  $R^2$  Ajustado)

```
# Revisar r cuadrado ajustado pero solo de un modelo
RsquareAdj(Modelo_Full)

# Selección paso a paso ajustada por r cuadrado ajustado
ordiR2step(Modelo_Nulo, Modelo_Full, trace=FALSE)
```

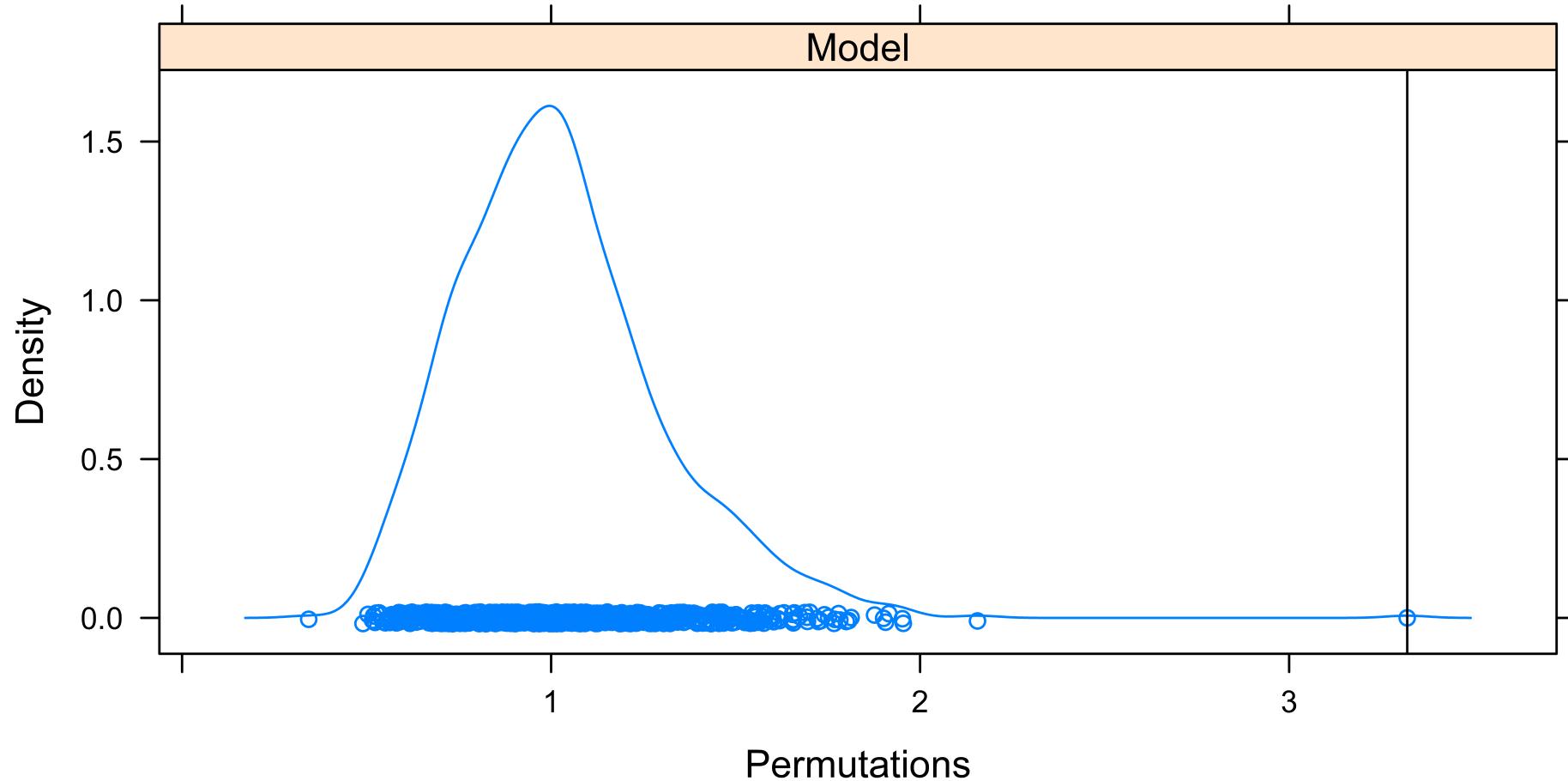


```
permustats(anova(RDA_vare)) |> summary()

##  
##      statistic    SES   mean lower median upper Pr(perm)  
## Model     3.3201 7.6986 1.0337      0.9927 1.5882    0.001 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## (Interval (Upper - Lower) = 0.95)
```



```
permustats(anova(RDA_vare)) |> densityplot()
```

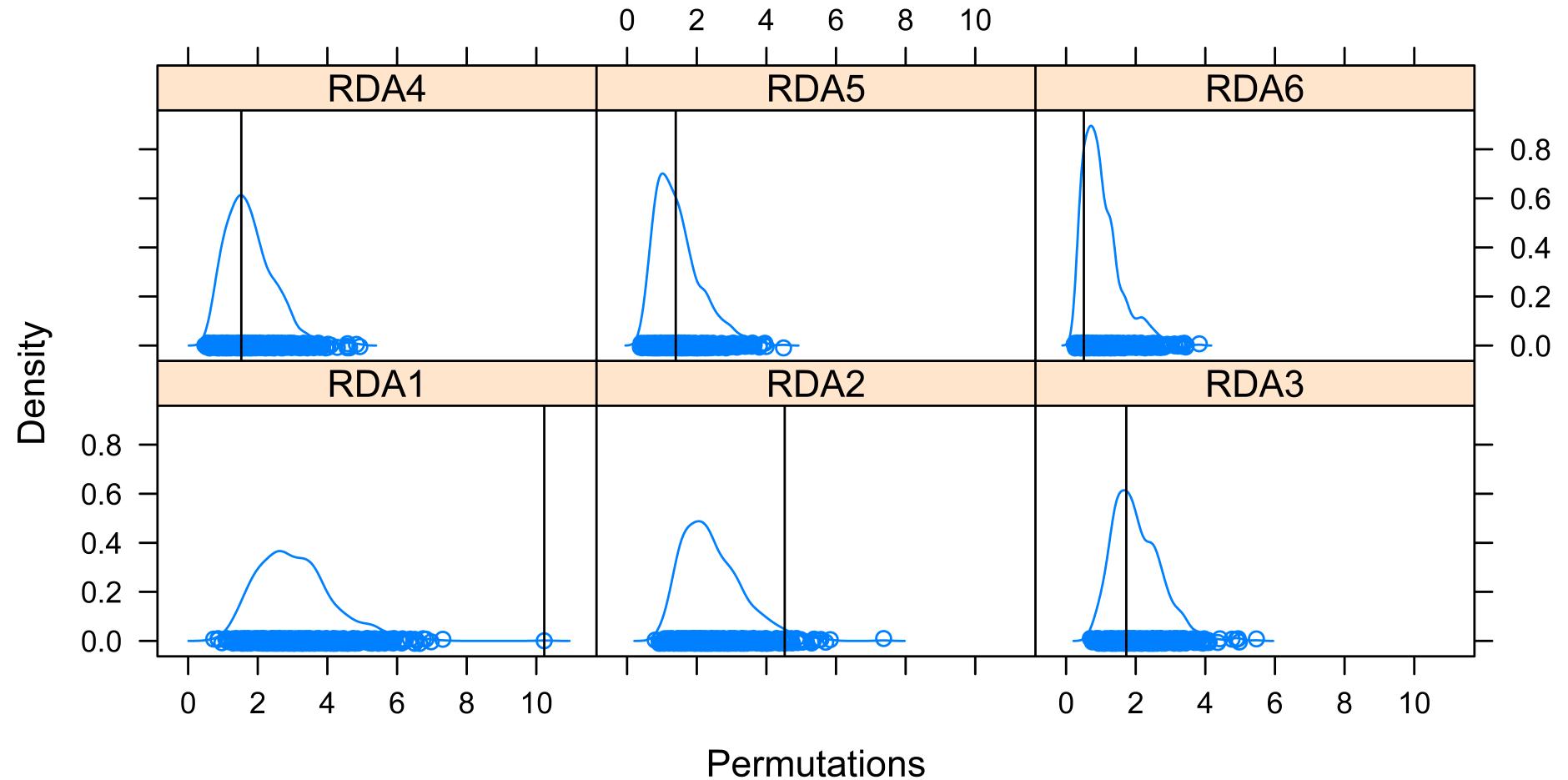


```
permustats(anova(RDA_vare, by="axis")) |> summary()
```

```
##  
##      statistic      SES   mean lower median upper Pr(perm)  
## RDA1    10.2293  5.9564 3.1164        2.9010  5.3902  0.001 ***  
## RDA2     4.5257  2.4354 2.3981        2.2512  4.0340  0.026 *  
## RDA3     1.7278 -0.3790 1.9945        1.8536  3.3135  0.588  
## RDA4     1.5279 -0.3411 1.7589        1.6172  3.0974  0.565  
## RDA5     1.3985 -0.0381 1.4227        1.3001  2.6939  0.436  
## RDA6     0.5112 -0.8799 1.0024        0.8886  2.1184  0.840  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## (Interval (Upper - Lower) = 0.95)
```



```
permustats(anova(RDA_vare, by="axis")) |> densityplot()
```

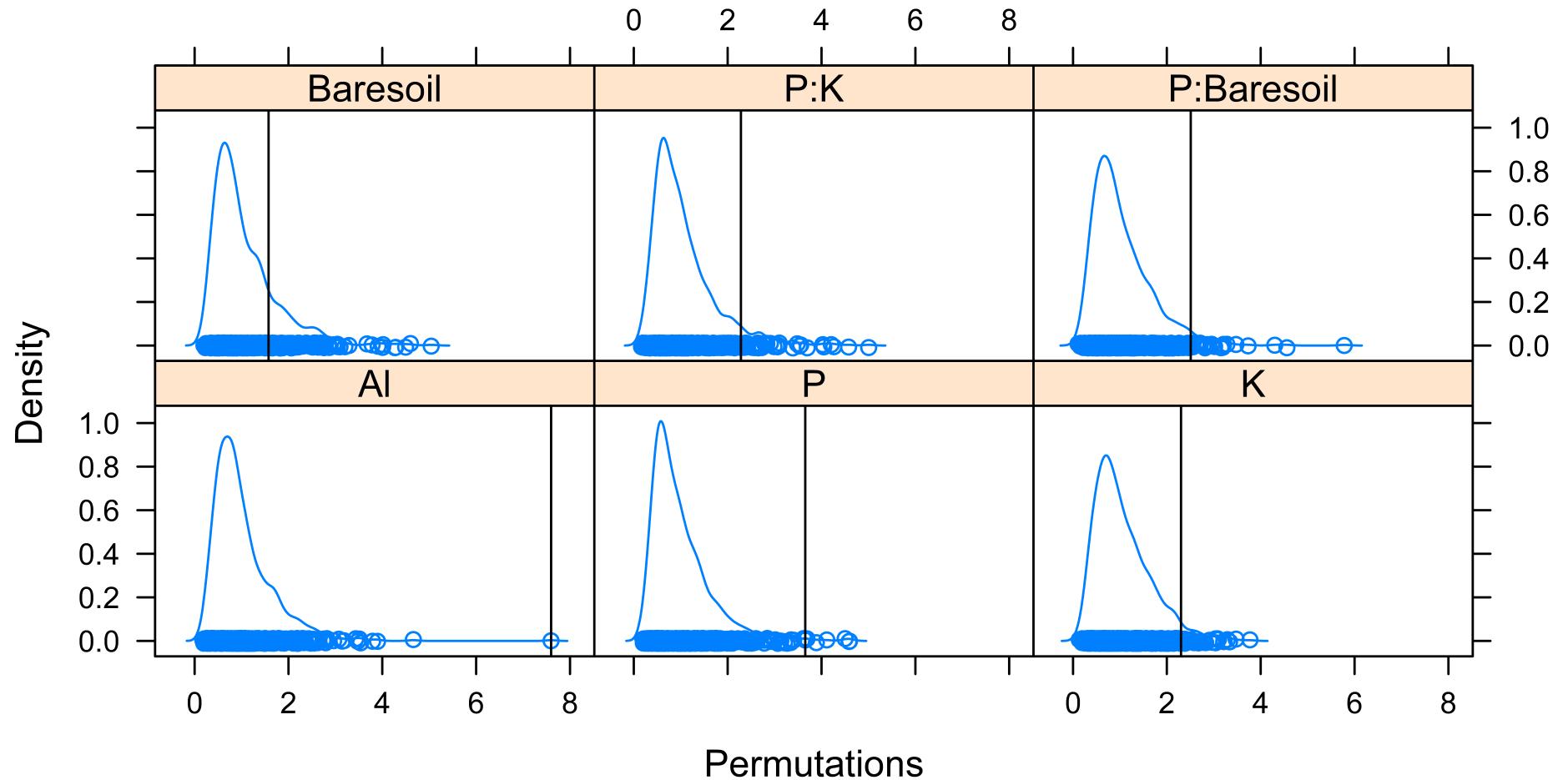


```
permustats(anova(RDA_vare, by="terms")) |> summary()
```

```
##  
##          statistic      SES   mean lower median upper Pr(perm)  
## Al        7.5980 10.3327 1.0235       0.8457 2.2821  0.001 ***  
## P         3.6535  3.9092 1.0650       0.8692 2.3554  0.004 **  
## K         2.3019  2.3279 1.0094       0.9160 2.0225  0.026 *  
## Baresoil  1.5761  0.8768 1.0137       0.8355 2.3487  0.144  
## P:K       2.2824  2.1308 1.0153       0.8438 2.1936  0.046 *  
## P:Baresoil 2.5084  2.6540 1.0007       0.8574 2.1265  0.023 *  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## (Interval (Upper - Lower) = 0.95)
```



```
permustats(anova(RDA_vare, by="terms")) |> densityplot()
```



# Gracias por su atención

