

Capítulo 3 - Association Measures

Melina Leite

24 de novembro de 2015

Estudo capítulo 3 livro Borcard et al. 2011. Numerical Ecology with R. Script provido pelo material da internet do livro.

Inclusões de informações do livro Legendre & Legendre 2012.

Métodos de análise multivariada se utilizam das medidas de associação (comparação de todos os pares de objetos possíveis), que podem ser implícitas em certas técnicas (como a distância euclidiana no PCA e K-means e distância qui-quadrado da CA) ou explícitas. Matriz de associação, é formada pelos índices de (dis)similaridade ou distância

As principais categorias de medidas de associação

Modo Q e R

Quando pares de **objetos** são comparados, a análise é em **modo Q**. No Q mode a medida de associação é a distância (dissimilaridade) ou a similaridade entre pares de objetos.

Quando pares de **descritores** são comparados, a análise é em **modo R**. No R mode, usa se uma medida de dependência entre variáveis, com covariância ou correlação.

Coefficientes symmetrical ou assymetrical em modo Q: problema do duplo zero

Em matrizes de espécie por local, o valor de zero nos locais é diferente do valor zero em matrizes de variáveis ambientais por local, por exemplo. A presença de uma espécie em dois locais geralmente implica que estes locais provém um grupo mínimo de condições permitindo as espécies sobreviver; essas condições são a dimensão do seu nicho ecológico. Porém a ausência de uma espécie em um local pode ocorrer devido à muitas causas e não necessariamente porque o local não é adequado para aquela espécie (pode ser limitação de dispersão, competição, etc), e também por detecção imperfeita do método.

1. A ausência de uma dada espécie em dois locais não pode ser facilmente contada como uma indicação de semelhança entre esses locais porque esta dupla ausência pode ser devido a razões completamente diferentes.
2. O número de double zeros não interpretáveis numa matriz depende do número de espécies e aumenta muito com o número de espécies raras detectadas.

Temos então dois tipos de medidas de associação:

- symmetrical: os que consideram os double-zeros (como o simple matching; legendre & legendre 2012 pg 296 tabela de índices)
- assimetrical: os que não consideram (como o Jaccard, Bray-Curtis, Sorensen)

OBS: não confundir os termos com *symetric* e *assymetric* que é quando o coeficiente de associação entre n_1 e n_2 é diferente ou não de n_2 e n_1 .

Quando temos matrizes de espécies, os zeros podem ser problemáticos e daí um coeficiente que não considera os double zeros é melhor do que o que considera.

Questões anteriores à análise

Antes de qualquer análise, responda as seguintes perguntas:

- você está comparando objetos (Q-mode) ou variáveis (R-mode)?
- Você está lidando com dados de espécies (geralmente coeficientes assymetricals) ou outros tipos de variáveis (symetricals) ?
- seus dados são binários (coeficiente binário), quantitativos (coef. quantitativos) ou mistura dos dois ou outros tipos (ordinal, coefs especiais)?

A Natureza do estudo determina o tipo de estrutura ecológica a ser evidenciada através de uma matriz de associação, e conseqüentemente o tipo de medida de similaridade a ser usada

- As várias medidas disponíveis tem diferentes restrições matemáticas. Você pode precisar de medidas diferentes se vc vai fazer agrupamento ou ordenação, por causa das propriedades matemáticas específicas.

Q Mode: computando matriz de distância entre objetos

Pacotes usados nessas análises:

```
library(ade4)
library(vegan) # should be loaded after ade4 to avoid some conflicts
library(cluster)
library(FD)
```

No R todas as medidas de similaridade são convertidas em distâncias para computar uma matriz quadrada de classe `dist` na qual a diagonal é zero e pode ser ignorada. A formula de conversão varia com o pacote:

- em `stats`, `FD` e `vegana` conversão da similaridade S para dissimilaridade D é $D = 1 - S$.
- No `ade4`, é computado como $D = \sqrt{1 - S}$. Isso permite que alguns índices se tornem Euclidianos, uma propriedade geométrica útil em algumas análises (como principal coordinate analysis). Matrizes de distância que não são euclidianas em outros pacotes podem ser tornar euclidianas computando $D2 = \sqrt{D}$.
- No `cluster`, todas as medidas disponíveis são distâncias, não precisa de conversão.

Q mode: dados quantitativos de espécies

Dados providos:

```
# Import the data from CSV files
spe <- read.csv("./NumEcolR_Data/DoubsSpe.csv", row.names=1)
env <- read.csv("./NumEcolR_Data/DoubsEnv.csv", row.names=1)
spa <- read.csv("./NumEcolR_Data/DoubsSpa.csv", row.names=1)
# Remove empty site 8
spe <- spe[-8,]
env <- env[-8,]
spa <- spa[-8,]
```

Lembrar que estes dados são semi-quantitativos, mas serão tratados como quantitativos. E que o local 8 não tem espécie nenhuma (por isso foi retirado).

No geral, são usadas medidas de distância assimétricas para dados quantitativos de espécies:

Bray-curtis dissilimilarity (D14) - tb conhecido como o recíproco do Steinhaus similarity index (S17)

pode ser computado diretamente dos dados brutos, mas geralmente transforma em log antes, porque D14 dá a mesma importância para diferenças absolutas na abundância, independente da magnitude da abundância.

Chord distance (D3)

É uma distância euclidiana computada em vetores de locais normalizados para comprimento 1, essa normalização se chama chord transformation. Feita no **vegan** com a função **decostand()**, argumento **normalize**.

Hellinger distance (D17)

é uma distância euclidiana nos vetores de locais, onde as abundâncias são primeiro divididas pela abundância total do local, e depois tirada a raiz quadrada. Tem efeito de reduzir a importância de grandes abundâncias. a transformação também pode ser feita na função **decostand** argumento **hellinger**.

Chi-square distance (D21)

Fala que essa distância também pode ser apropriada para dados de abundância

```
# Bray-Curtis dissimilarity matrix on raw species data
spe.db <- vegdist(spe) # Bray-Curtis dissimilarity (default)
head(spe.db)
```

```
## [1] 0.6000000 0.6842105 0.7500000 0.8918919 0.7500000 0.6842105
```

```
# Bray-Curtis dissimilarity matrix on log-transformed abundances
spe.dbln <- vegdist(log1p(spe))
head(spe.dbln)
```

```
## [1] 0.5509095 0.6280761 0.7446012 0.8664653 0.7657643 0.6519273
```

```
# Chord distance matrix
spe.norm <- decostand(spe, "nor")
spe.dc <- dist(spe.norm)
head(spe.dc)
```

```
## [1] 0.7653669 0.9235374 1.0432236 1.2786132 1.1153793 0.8873116
```

```
# Hellinger distance matrix
spe.hel <- decostand(spe, "hel")
spe.dh <- dist(spe.hel)
head(spe.dh)
```

```
## [1] 0.8420247 0.9391305 1.0616631 1.2308244 1.1153793 0.9391305
```

Q-mode: dados de espécies binário (presença-ausência)

Índices de similaridade de Jaccard (S7)

para cada par de locais, o Jaccard é a razão entre o número de duplos 1s e o número de espécies, excluindo as espécies representadas pelos duplos 0s.

```
# Jaccard dissimilarity matrix using function vegdist()
spe.dj <- vegdist(spe, "jaccard", binary=TRUE)
head(spe.dj)
```

```
## [1] 0.6666667 0.7500000 0.8750000 0.9090909 0.9000000 0.8000000
```

```
head(sqrt(spe.dj))
```

```
## [1] 0.8164966 0.8660254 0.9354143 0.9534626 0.9486833 0.8944272
```

```
# Jaccard dissimilarity matrix using function dist()
spe.dj2 <- dist(spe, "binary")
head(spe.dj2)
```

```
## [1] 0.6666667 0.7500000 0.8750000 0.9090909 0.9000000 0.8000000
```

```
# Jaccard dissimilarity matrix using function dist.binary()
spe.dj3 <- dist.binary(spe, method=1)
head(spe.dj3)
```

```
## [1] 0.8164966 0.8660254 0.9354143 0.9534626 0.9486833 0.8944272
```

lembrar que os índices com as diferentes funções dão valores diferentes porque calculam de forma diferente

Sorensen (S8)

Dá mais peso ao número de duplos 1s e é seu recíproco (complemento para 1) é equivalente a distância de bray-curtis computada para dados de presença-ausência.

```
# Sorensen dissimilarity matrix using function dist.binary()
spe.ds <- dist.binary(spe, method=5)
head(spe.ds)
```

```
## [1] 0.7071068 0.7745967 0.8819171 0.9128709 0.9045340 0.8164966
```

```
# Sorensen dissimilarity matrix using function vegdist()
spe.ds2 <- vegdist(spe, binary=TRUE)
head(spe.ds2)
```

```
## [1] 0.5000000 0.6000000 0.7777778 0.8333333 0.8181818 0.6666667
```

```
head(sqrt(spe.ds2))
```

```
## [1] 0.7071068 0.7745967 0.8819171 0.9128709 0.9045340 0.8164966
```

Ochiai (S14)

Também apropriado para dados de presença-ausência, está relacionado às distâncias de chord e hellinger. Estas distâncias, assim como a de qui-quadrado são boas para dados de presença-ausência.

```
# Ochiai dissimilarity matrix  
spe.och <- dist.binary(spe, method=7)  
head(spe.och)
```

```
## [1] 0.6501152 0.7071068 0.8040190 0.8357563 0.8269052 0.7434961
```

Visualização gráfica das matrizes de distância/dissimilaridade:

Usando o pacote `gclus` para propósito de visualização dentro da função `coldiss()` (que é provida pelo livro).

Daí as cores vão indicar os valores de similaridade/dissimilaridade entre os locais, a função `coldiss()` também pode ser vista ordenada para ver as que mais se relacionam.

É bom para comparar as diferenças entre os índices.

Cores:

- magenta: dissimilaridade perto de 0 (similaridade máxima)
- ciano: dissimilaridade perto de 1 (similaridade mínima)

```
# The gclus package is required and may be called now, although  
# it is called internally by coldiss()  
library(gclus)
```

```
##
```

```
## Attaching package: 'gclus'
```

```
## The following object is masked from 'package:magic':
```

```
##
```

```
## diag.off
```

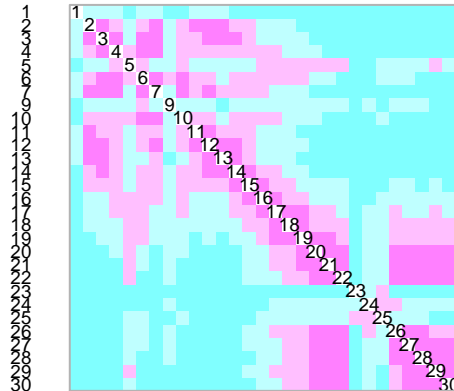
```
# Colour plots (also called heat maps, or trellis diagrams in the data analysis literature) using the c  
# *****
```

```
# Source the coldiss() function  
source("../NumEcolR_functions/coldiss.R") # If necessary, add the path to the file
```

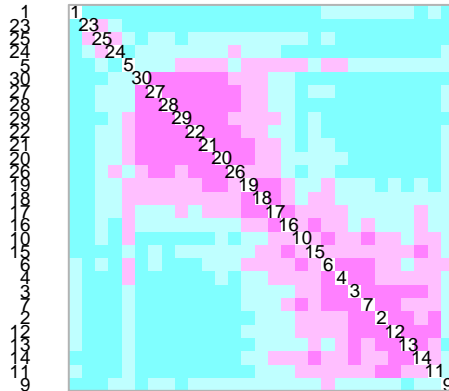
Compare os dois plots do **bray-curtis** com os dados brutos e log-transformados. Na matriz dos dados não transformados, as pequenas diferenças nas abundâncias de espécies tem a mesma importância que as grandes diferenças com poucos indivíduos.

```
# Bray-Curtis dissimilarity matrix (on raw data)
# 4 colours with equal-length intervals (useful for comparisons)
#quartz(title="Bray-Curtis (raw data)",10,5)
coldiss(spe.db, byrank=FALSE, diag=TRUE)
```

Dissimilarity Matrix

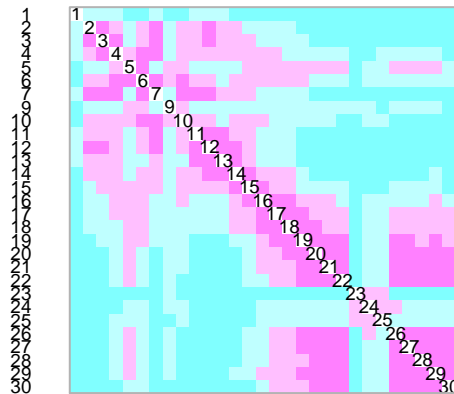


Ordered Dissimilarity Matrix

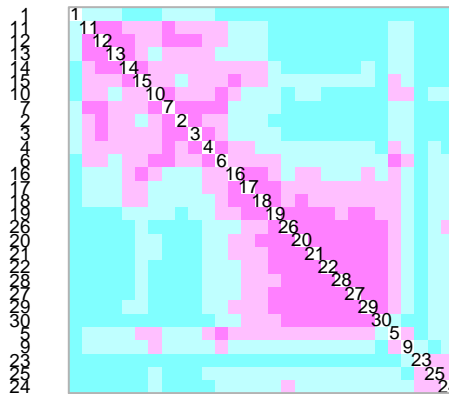


```
# Same but on log-transformed data
#quartz(title="Bray-Curtis [ln(y+1) data]",10,5)
coldiss(spe.dbln, byrank=FALSE, diag=TRUE)
```

Dissimilarity Matrix



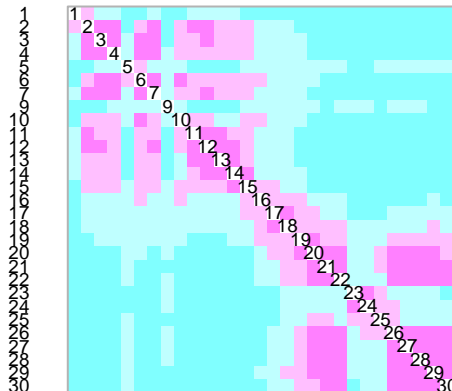
Ordered Dissimilarity Matrix



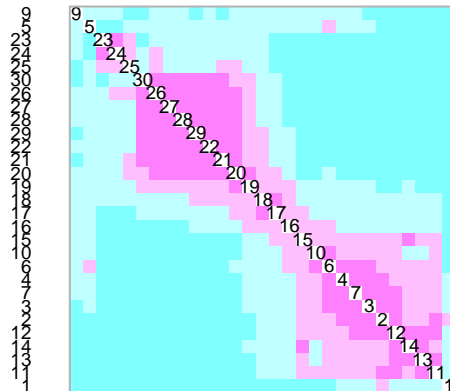
Comparando com outras distâncias quantitativas:

```
# Chord distance matrix
#quartz(title="Chord",10,5)
coldiss(spe.dc, byrank=FALSE, diag=TRUE)
```

Dissimilarity Matrix

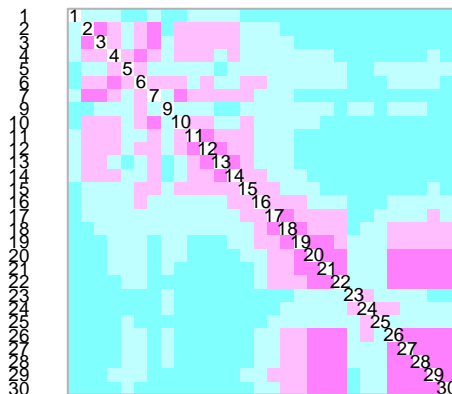


Ordered Dissimilarity Matrix

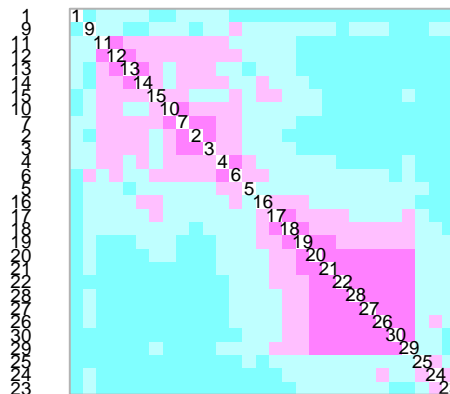


```
# Hellinger distance matrix
#quartz(title="Hellinger",10,5)
coldiss(spe.dh, byrank=FALSE, diag=TRUE)
```

Dissimilarity Matrix



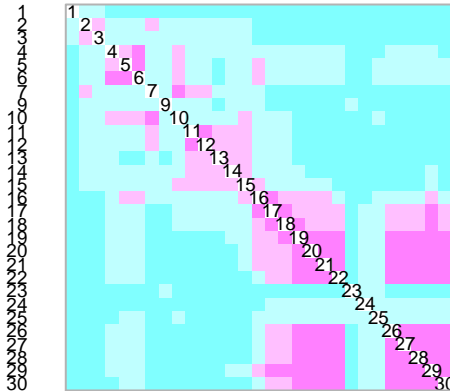
Ordered Dissimilarity Matrix



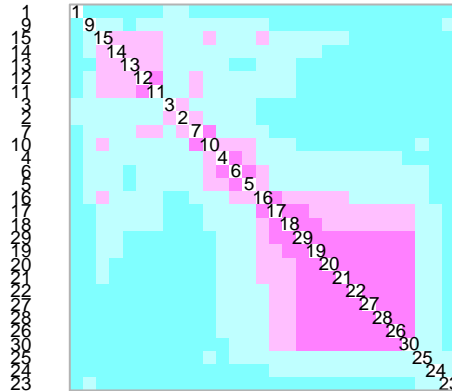
Compare agora com o índice de jaccard para dados binários. Isso influencia os resultados?

```
# Jaccard distance matrix
#quartz(title="Jaccard",10,5)
coldiss(spe.dj, byrank=FALSE, diag=TRUE)
```

Dissimilarity Matrix



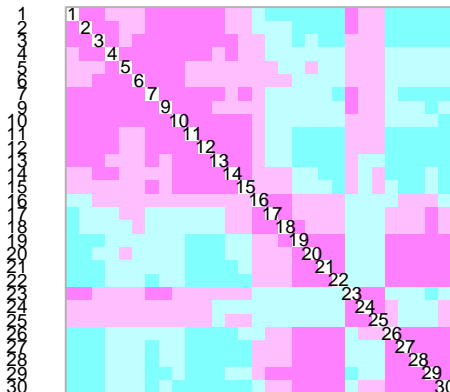
Ordered Dissimilarity Matrix



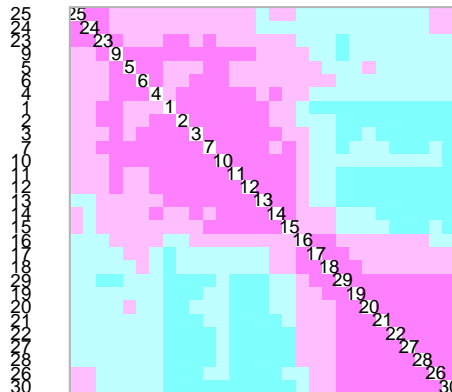
O índice de **imple matching** inclui o double zero. Se nós assumirmos que as ausências de espécies são devido principalmente às variáveis ambientais e que as espécies seriam capazes de dispersar para todos os locais, podemos usá-lo:

```
# Simple matching dissimilarity
# (called the Sokal and Michener index in ade4)
spe.s1 <- dist.binary(spe, method=2)
#quartz(title="S1 on species data",10,5)
coldiss(spe.s1^2, byrank=FALSE, diag=TRUE)
```

Dissimilarity Matrix



Ordered Dissimilarity Matrix



Q Mode: dados quantitativos ambientais

Nesse caso há uma clara interpretação do **double zero**. A distância simétrica mais usual é a **Euclidiana** (D1). Não tem limite superior e o valor é altamente influenciado pela escala de cada descritor. Então, pra usar tem que padronizar (pelo menos), geralmente usa-se z-scores. Mesmo com uma matriz dimensionalmente homogênea, muitos vão padronizar para dar o mesmo peso para toda as variáveis.

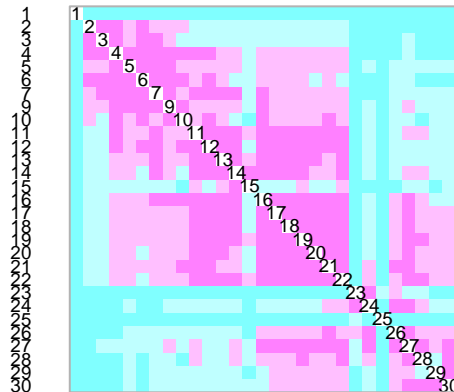
Computando a matriz de distância euclidiana dos dados ambientais padronizados:

```
# Remove the 'das' variable from the env dataset
env2 <- env[,-1] # é variavel espacial e nao ambiental
```

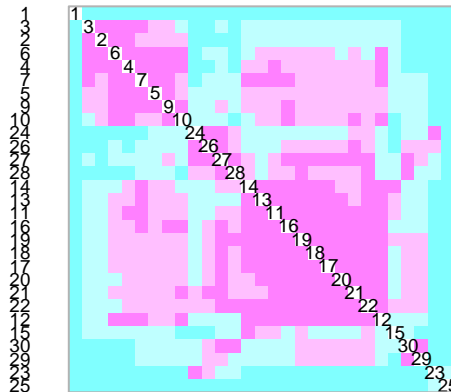


```
# Euclidean distance matrix of the standardized env2 data frame
env.de <- dist(scale(env2))
#quartz(title="Environment",10,5)
coldiss(env.de, diag=TRUE)
```

Dissimilarity Matrix



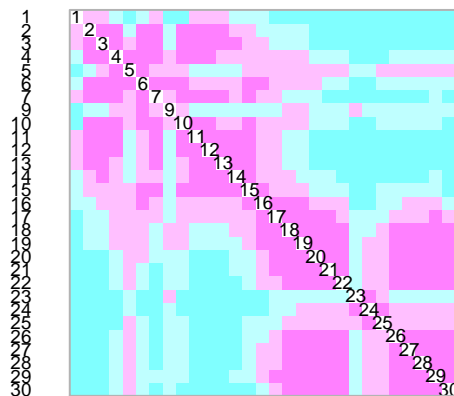
Ordered Dissimilarity Matrix



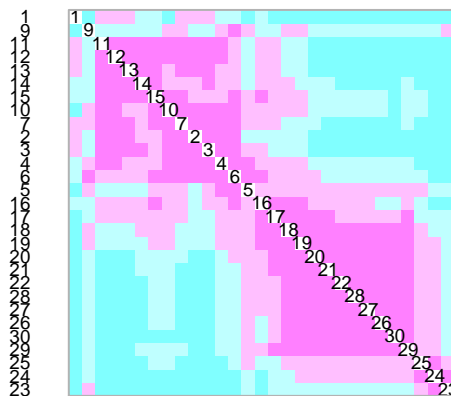
Fazendo com a distância de hellinger para comparar:

```
# Hellinger distance matrix of the species data (equal-sized categories)
#quartz(title="Species",10,5)
coldiss(spe.dh, diag=TRUE)
```

Dissimilarity Matrix



Ordered Dissimilarity Matrix

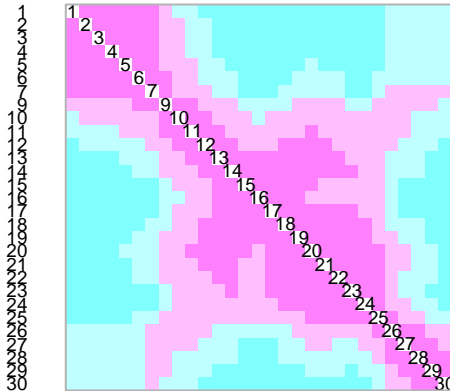


A distância euclidiana é apropriada para computar matrizes de distâncias geográficas (1 ou 2 dimensões). Mas se as coordenadas estiverem em lat-long tem que transformar antes (usar função `geoXY` do pacote `SoDA`).

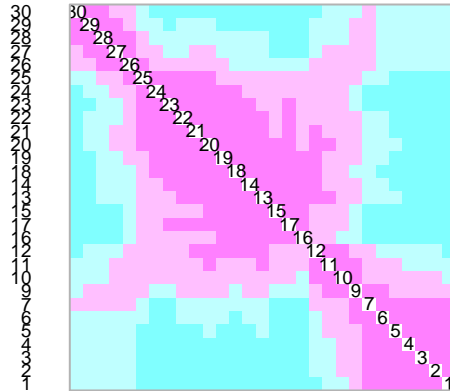
Abaixo temos as distâncias calculadas para 2D e para 1D (usando a distância pelo curso do rio):

```
# Euclidean distance matrix on spatial coordinates (2D)
spa.de <- dist(spa)
#quartz(title="x-y",10,5)
coldiss(spa.de, diag=TRUE)
```

Dissimilarity Matrix

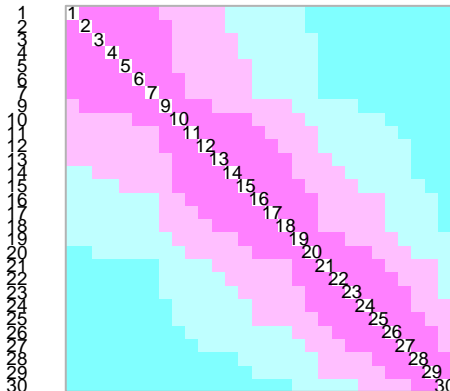


Ordered Dissimilarity Matrix

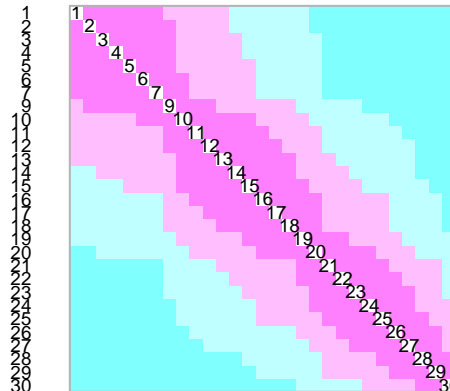


```
# Euclidean distance matrix on distance from the source (1D)
das.df <- as.data.frame(env$das, row.names=row.names(env))
riv.de <- dist(das.df)
#quartz(title="Distance from source",10,5)
coldiss(riv.de, diag=TRUE)
```

Dissimilarity Matrix



Ordered Dissimilarity Matrix



Q Mode: dados quantitativos ambientais binários

A medida de similaridade symmetrical mais simples para dados binários ambientais é o **simple matching**.

Criando dados para calcular o índice:

```
# Compute five binary variables with 30 objects each. Each variable
# has a predefined number of 0 and 1
# Variable 1: 10 x 1 and 20 x 0; the order is randomized
var1 <- sample(c(rep(1,10), rep(0,20)))
# Variable 2: 15 x 0 and 15 x 1, one block each
var2 <- c(rep(0,15), rep(1,15))
# Variable 3: alternation of 3 x 1 and 3 x 0 up to 30 objects
var3 <- rep(c(1,1,1,0,0,0),5)
# Variable 4: alternation of 5 x 1 and 10 x 0 up to 30 objects
```

```

var4 <- rep(c(rep(1,5), rep(0,10)), 2)
# Variable 5: 16 objects with randomized distribution of 7 x 1
# and 9 x 0, followed by 4 x 0 and 10 x 1
var5.1 <- sample(c(rep(1,7), rep(0,9)))
var5.2 <- c(rep(0,4), rep(1,10))
var5 <- c(var5.1, var5.2)

# Variables 1 to 5 are put into a data frame
dat <- data.frame(var1, var2, var3, var4, var5)
dim(dat)

```

```
## [1] 30 5
```

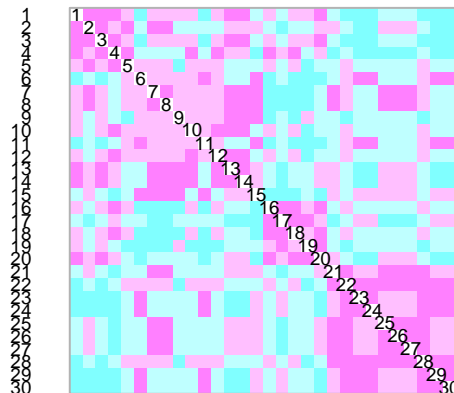
Fazendo o simple matching para essa matriz:

```

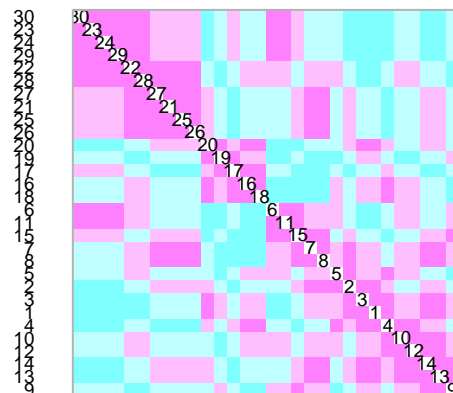
# (called Sokal and Michener index in ade4)
dat.s1 <- dist.binary(dat, method=2)
#quartz(title="S1 on fictitious data",10,5)
coldiss(dat.s1, diag=TRUE)

```

Dissimilarity Matrix



Ordered Dissimilarity Matrix



Q Mode: tipos mistos, incluindo variáveis categóricas (qualitativas multiclases)

Medida de associação que consegue lidar com dados nominais é o **Gower's similarity** (S15). Esse coeficiente foi criado para lidar com dados contendo vários tipos matemáticos, cada variável recebendo um tratamento correspondente à sua categoria. A dissimilaridade final entre dois objetos é obtida pela média da (dis)similaridade computada para todas as variáveis separadamente.

Para dados como fatores, aplica a regra do simple matching: para cada par de objetos a similaridade é 1 para aquela variável que tem o mesmo nível nos dois objetos e zero se os níveis são diferentes.

no R:

```

# Fictitious data for Gower (S15) index
# Random normal deviates with zero mean and unit standard deviation
var.g1 <- rnorm(30,0,1)
# Random uniform deviates from 0 to 5

```

```

var.g2 <- runif(30,0,5)
# Factor with 3 levels (10 objects each)
var.g3 <- gl(3,10)
# Factor with 2 levels, orthogonal to var.g3
var.g4 <- gl(2,5,30)
dat2 <- data.frame(var.g1,var.g2,var.g3,var.g4)
summary(dat2)

```

```

##      var.g1      var.g2      var.g3 var.g4
## Min.   :-2.7215  Min.   :0.008793  1:10  1:15
## 1st Qu.: -0.9672  1st Qu.:0.767409  2:10  2:15
## Median :  0.4100  Median :2.235324  3:10
## Mean   :  0.1040  Mean   :2.232511
## 3rd Qu.:  1.1495  3rd Qu.:3.348812
## Max.    :  2.0894  Max.    :4.803659

```

```

# Computation of a matrix of Gower dissimilarity using function daisy() pacote cluster
# evitar usar vegdist(, method='gower') - só apropriado para dados quantitativos e binários.

# Complete data matrix (4 variables)
dat2.S15 <- daisy(dat2, "gower")
range(dat2.S15)

```

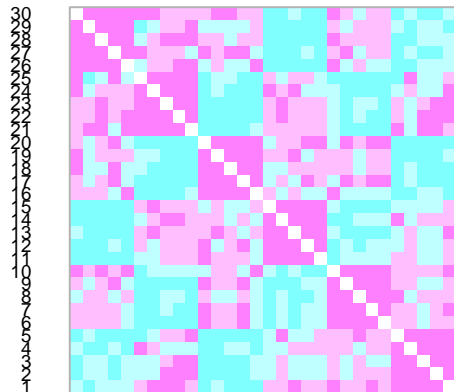
```
## [1] 0.0110912 0.8530021
```

```

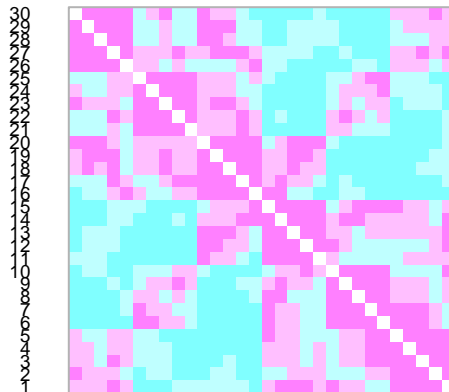
#quartz(title="S15 on fictitious data - daisy",10,5)
coldiss(dat2.S15, diag=TRUE)

```

Dissimilarity Matrix



Ordered Dissimilarity Matrix

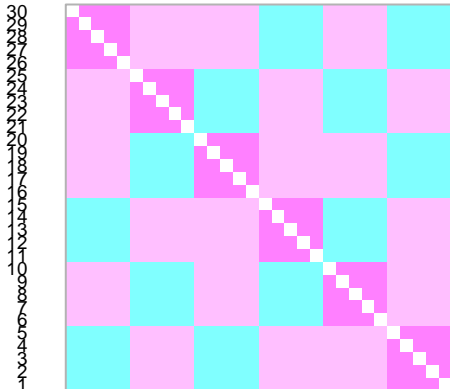


```

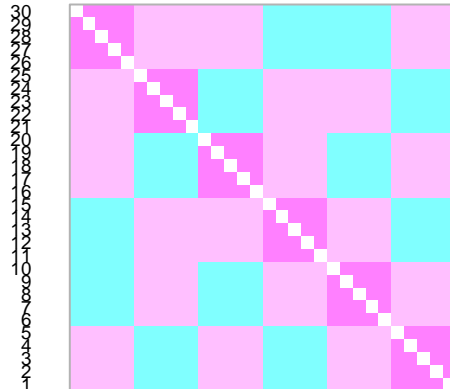
# Data matrix with the two orthogonal factors only
dat2partial.S15 <- daisy(dat2[,3:4], "gower")
#quartz(title="S15 on fictitious data, 2 factors - daisy",10,5)
coldiss(dat2partial.S15, diag=TRUE)

```

Dissimilarity Matrix



Ordered Dissimilarity Matrix



```
# What are the dissimilarity values in the dat2partial.S15 matrix?
levels(factor(dat2partial.S15))
```

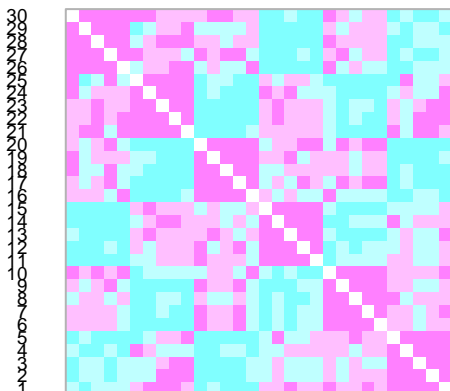
```
## [1] "0" "0.5" "1"
```

```
# Computation of a matrix of Gower dissimilarity using function gowdis()
# of package FD
library(FD) # If not already loaded
?gowdis
dat2.S15.2 <- gowdis(dat2)
range(dat2.S15.2)
```

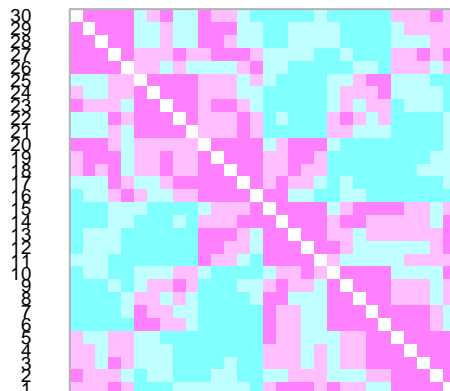
```
## [1] 0.0110912 0.8530021
```

```
#quartz(title="S15 on fictitious data - gowdis",10,5)
coldiss(dat2.S15.2, diag=TRUE)
```

Dissimilarity Matrix

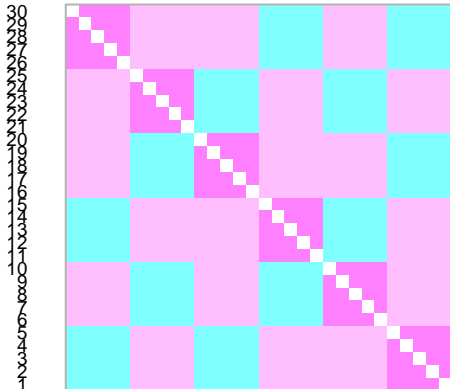


Ordered Dissimilarity Matrix

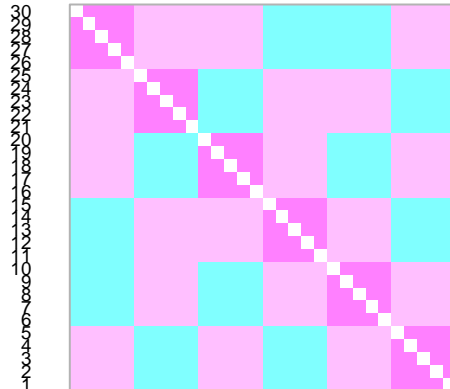


```
# Data matrix with the two orthogonal factors only
dat2partial.S15.2 <- gowdis(dat2[,3:4])
#quartz(title="S15 on fictitious data, 2 factors - gowdis",10,5)
coldiss(dat2partial.S15.2, diag=TRUE)
```

Dissimilarity Matrix



Ordered Dissimilarity Matrix



```
# What are the dissimilarity values in the dat2partial.S15.2 matrix?
levels(factor(dat2partial.S15.2))
```

```
## [1] "0" "0.5" "1"
```

R mode: Computando matrizes de dependência entre variáveis

Coefficientes tipo correlação devem ser usados para comparar variáveis no R mode. Usa-se **Pearson** (dados paramétricos quantitativos), **Spearman** ou **Kendall** para dados não paramétricos, quantitativos ou semi-quantitativos, e estatísticas de contingência para comparações de variáveis qualitativas (estatística qui-quadrado, etc)

Também se usa Jaccard, Sorensen e Ochiai para dados binários de matriz de espécies.

R mode: Dados de abundância de espécies

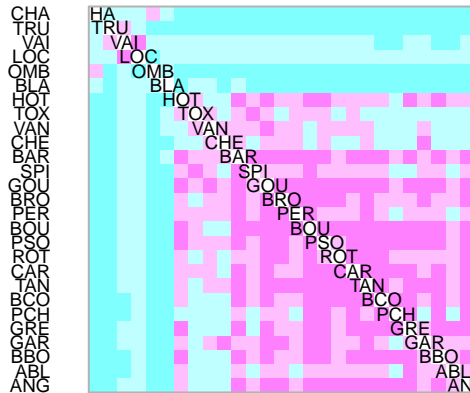
Para comparar a distribuição de espécies no espaço ou tempo. Geralmente recomenda-se pré-transformação dos dados (ver seção abaixo).

No exemplo usamos na verdade a distância de qui-quadrado na matriz R mode

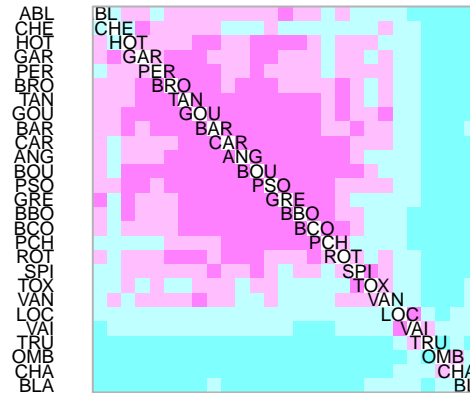
```
# Transpose matrix of species abundances
spe.t <- t(spe)

# Chi-square pre-transformation followed by Euclidean distance
spe.t.chi <- decostand(spe.t, "chi.square")
spe.t.D16 <- dist(spe.t.chi)
#quartz(title="D16 on fish species (R-mode)",10,5)
coldiss(spe.t.D16, diag=TRUE)
```

Dissimilarity Matrix



Ordered Dissimilarity Matrix

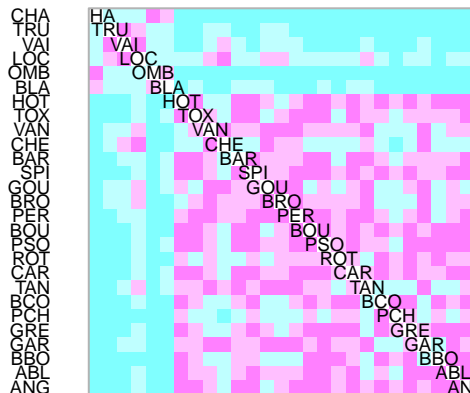


R mode: dados presença-ausência espécies

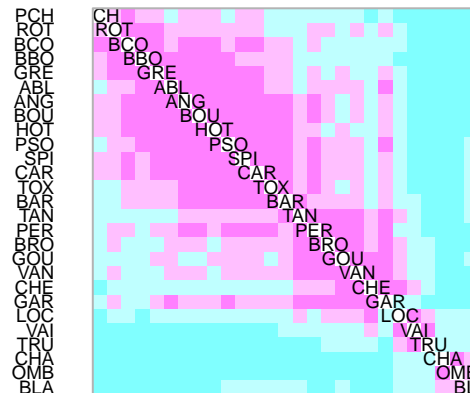
Para dados binários, Jaccard, Sorensen e Ochiai também podem ser usados no modo R:

```
# Jaccard index on fish presence-absence
spe.t.S7 <- vegdist(spe.t, "jaccard", binary=TRUE)
#quartz(title="S7 on fish species (R-mode)",10,5)
coldiss(spe.t.S7, diag=TRUE)
```

Dissimilarity Matrix



Ordered Dissimilarity Matrix



R mode: dados quantitativos e ordinais (não espécies)

Usa-se coeficiente de correlação de **Pearson** geralmente. Tomar cuidado que pearson é um coeficiente linear, e não pesca dados monotônicos não lineares. Daí pode usar o **Speraman** ou **Kendall**:

```
# Pearson r linear correlation among environmental variables
env.pearson <- cor(env) # default method = "pearson"
round(env.pearson, 2)
```

```
##      das  alt  pen  deb  pH  dur  pho  nit  amm  oxy  dbo
```

```
## das 1.00 -0.94 -0.39 0.95 0.02 0.73 0.47 0.74 0.41 -0.57 0.43
## alt -0.94 1.00 0.46 -0.86 -0.05 -0.79 -0.44 -0.75 -0.38 0.42 -0.38
## pen -0.39 0.46 1.00 -0.36 -0.22 -0.53 -0.20 -0.31 -0.17 0.31 -0.17
## deb 0.95 -0.86 -0.36 1.00 0.03 0.74 0.38 0.59 0.29 -0.42 0.30
## pH 0.02 -0.05 -0.22 0.03 1.00 0.08 -0.08 -0.04 -0.12 0.19 -0.16
## dur 0.73 -0.79 -0.53 0.74 0.08 1.00 0.37 0.53 0.30 -0.37 0.34
## pho 0.47 -0.44 -0.20 0.38 -0.08 0.37 1.00 0.80 0.97 -0.76 0.91
## nit 0.74 -0.75 -0.31 0.59 -0.04 0.53 0.80 1.00 0.80 -0.69 0.68
## amm 0.41 -0.38 -0.17 0.29 -0.12 0.30 0.97 0.80 1.00 -0.75 0.90
## oxy -0.57 0.42 0.31 -0.42 0.19 -0.37 -0.76 -0.69 -0.75 1.00 -0.84
## dbo 0.43 -0.38 -0.17 0.30 -0.16 0.34 0.91 0.68 0.90 -0.84 1.00
```

```
# Reorder the variables prior to plotting
```

```
env.o <- order.single(env.pearson)
```

```
# pairs: a function to plot a matrix of bivariate scatter diagrams and correlation coefficients.
```

```
# Correlations are given in the upper panel (with significance levels)
```

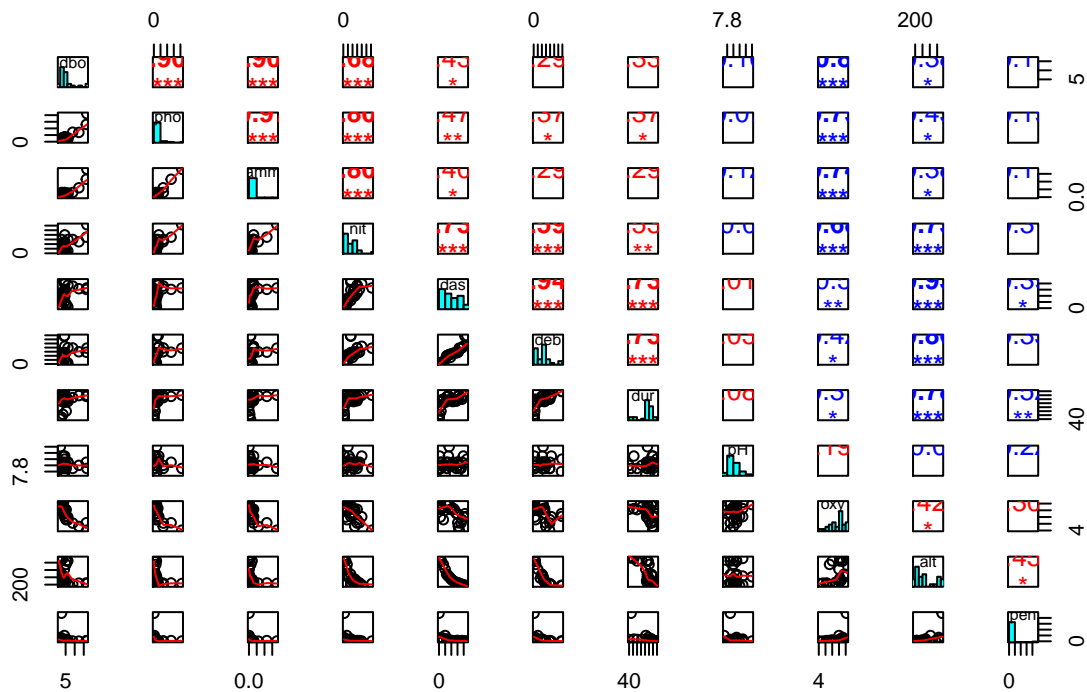
```
source("./NumEcolR_functions/panelutils.R") # If necessary give path
```

```
#quartz(title="Linear correlation matrix",10,10)
```

```
op <- par(mfrow=c(1,1), pty="s")
```

```
pairs(env[,env.o], lower.panel=panel.smooth, upper.panel = panel.cor,
      diag.panel=panel.hist, main="Pearson Correlation Matrix")
```

Pearson Correlation Matrix



```
par(op)
```

```
# Kendall tau rank correlation among environmental variables
```

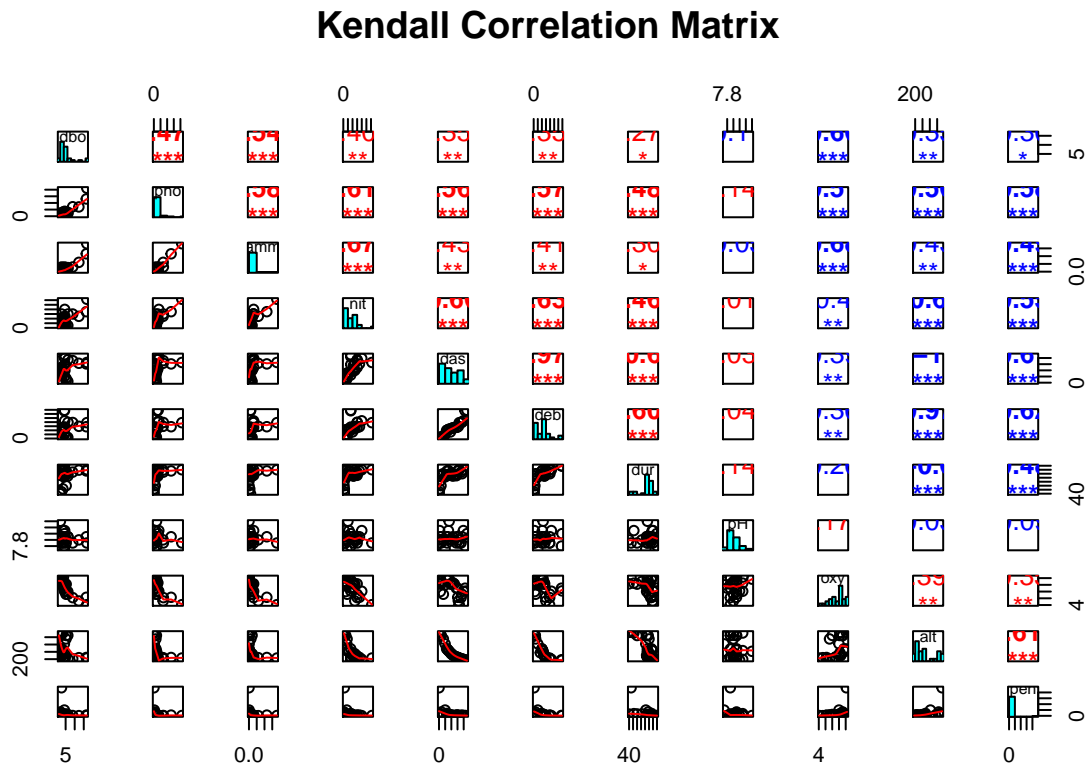
```
env.ken <- cor(env, method="kendall")
```

```
env.o <- order.single(env.ken)
```

```
#quartz(title="Rank correlation matrix",10,10)
```



```
op <- par(mfrow=c(1,1), pty="s")
pairs(env[,env.o], lower.panel=panel.smooth, upper.panel = panel.cor,
      method="kendall", diag.panel=panel.hist, main="Kendall Correlation Matrix")
```



```
par(op)
```

Veja os plots das relações, devemos usar kendall ou pearson?

R mode: dados binários (não espécies)

Computar uma matriz de **Pearson's r** .

Pré-transformações dos dados de espécies

Modelos lineares que usam distância euclidiana (anova, kmeans, rda, pca) não são bons pra dados de abundâncias. Mas atualmente temos medidas de associação assymetrical que podem ser obtidos em 2 passos: transforma os dados brutos e depois calcula a distância euclidiana.

Usar função `decostand()` com os diferente métodos de transformação: “total”(perfil de abundâncias relativas), “normalize” (transformação de chord), “chi.square”, “hellinger” (diminui bastante os valores de abundância muito altos)...

Todas as transformações expressam os dados como abundâncias relativas por locais (perfil de locais), isso remove dos dados as abundâncias totais por local (daí não dá pra ver produtividade por exemplo).

(ver capítulo 2 com essas transformações).

Conclusão

Recomenda que os coeficientes de similaridade para presença-absência devem ser transformados em distâncias $\sqrt{1 - S}$, para evitar a produção de autovalores negativos e imaginários nas análises de coordenadas principais. Essa transformação é feita automaticamente no **ade4** mas não no **vegan** .