

Revisão Geral.

Jaques E. Zanon

junho/2021

Comandos Básicos

```
# Atribuindo resultados a objetos:

# R como calculadora

resultado <- 4+1
resultado_1 <- 2*4*10

# Concatenar c()

resultado <- c(1,2,3,4,5) # concatenar
resultado <- c(14,21,32,49,55)

# Acessando valores dentro do vetor:

resultado[1:5]

obj <- resultado[c(1,3)]

c(resultado,resultado,resultado)

# Concatenado texto sempre entre aspas

obj_3 <- c("chuva","seca","chuva","seca","chuva","ano1","ano2")

obj_3[1:4]

# Concatenando colunas cbind()

cbind(obj_2,obj_3)

mat<-cbind(obj_2,obj_2)

# Acessando linhas e colunas:

mat[c(1,3),] # linhas

mat[,]

mat[,c(1,3)] # colunas
```

```

mat[c(1,3),2]

# Nomes das colunas:

colnames(mat) <- c("resposta1", "resposra2")
colnames(mat)

cbind(mat,mat)

# Concatenando linhas rbind()

rbind(mat[1,],mat[2,])

## Sempre que estiver em dúvida acerca da função.

help("funcao")

```

Importando dados & Principios de Analise exploratoria.

```

# Importando usando o exemplo dado em aula "spe.csv".
# Primeiro o nome do arquivo em "csv".
# header= TRUE para utilizar a primeira linha como nomes das varaveis/colunas.
# Se quiser a primeira coluna como nome das linhas row.names=1.
# sep = ";" delimita o separados para as colunas, pode ser também sep = ",".
# dec = "," é o separador decimal dos valores.

spe <- read.csv("DoubSpe.csv", header = TRUE, row.names = 1, sep = ";", dec=",")
head(spe)

# Agora com arquivo de texto
# Os argumentos são semelhantes aos anteriores.

df<-read.table("nome_do_arquivo.txt",header=T)

# Agora para arquivo formato excel ".xlsx"
# Especificar a aba pelo nome
# O restante dos argumentops são semelhantes ao "read.csv"

my_data <- read_excel("my_file.xlsx", sheet = "data")

# Ou pelo numero.
my_data <- read_excel("my_file.xlsx", sheet = 1)

#####
*****  Análise exploratória  *****

# Sempre que importar utilize o comando head() para visualizar as primiras linhas.
head()

```

```
## Quantas espécies?/Número de colunas.
ncol()

## Quantas obs?/Número de linhas.
nrow()

## Locais com zero espécies?/ Soma das linhas.
rowSums()

## Tamanho das populações de sp./Soma das colunas.
colSums()

## Verificar se existem vaLORES faltantes "NA".
is.na()

## Deletar linhas com valores faltantes.
na.omit()

## Lembre-se uma matriz possui apenas valores numericos e
## um data frame possui numeros e fatores (text).

# para transformar em matriz e dataframe.
as.matrix(); as.data.frame()
```

Estatística univariada

```
# Em geral, a maioria das análises é feita especificando-se o modelo que se deseja testar.
# No R, para especificar um modelo é preciso usar a notação de formulas.
# A regressão linear simples é utilizada para analisar relações entre variáveis contínuas.

resultado<-lm(Y ~ X) # a notação ~ significa fórmula.
resultado

# Para inspecionar os residuos.

residuals()

#Veja que o resultado da regressão mostra os coeficientes:
# a (intercepto) e b (inclinação) da regressão.

summary(resultado)

## Regressão Múltipla
# A regressão linear com uma única variável preditora X pode ser facilmente estendido a duas ou mais
# variáveis preditoras (regressão múltipla).

reg.mult<-lm(Y ~ X1 + X2)
summary(reg.mult)

### Análise de Variância (Anova) ###
```

```
resu.aov<-aov(Y~X)
summary(resu)
```

Instalando e carregando pacotes

```
# Bem simples o primeiro comando para instalar e o segundo para carregar
install.packages("vegan")
library(vegan)

## Existe ainda uma opção caso não queira carregar o pacote mas utilizar determinada função:
# Mas o pacote precisa estar instalado.

vegan::rda() # exemplo para usar a função rda sem ter que carregar o pacote.
```

Medidas de associação.

```
library(vegan)

## Distâncias, Q mode

# Bray-Kurtis, lembre-se, a matriz só pode ter dados numéricos, ou seja, sem texto (apenas um passo).
spe.db <- vegdist(spe, method="bray") # method = "bray"

# Chord distance / Distância da corda (dois passos)
spe.norm <- decostand(spe, "standardize")
spe.dc <- vegdist(spe.norm, method="euclidian")

# Distância Hellinger (dois passos)
spe.hel <- decostand(spe, "hel") # ponderar sp muito abundante
spe.dh <- vegdist(spe.hel, method="euclidian")

## R mode

# Correlação
# Pearson r linear correlation among environmental variables

env.cor <- cor(env)
env.cor
as.dist(env.cor) # aqui para deixar como matriz de distância (triangular).
```

Agrupamento não supervisionado (K-means)-análise multivariada

```
# Instalando alguns pacotes
install.packages("cluster")
install.packages("factoextra")
install.packages("ggplot2")
```

```

library(cluster)      # clustering algorithms
library(factoextra)   # clustering algorithms & visualization
library(ggplot2)

# Aplicando K-means para dados transformados.
# O argumento centers se refere ao número de grupos a serem identificados.

spe.hel <- decostand(spe, method = "hel")
k2 <- kmeans(spe.hel, centers = 2, nstart = 25)
fviz_cluster(k2, data = spe.hel)

k3 <- kmeans(spe.hel, centers = 3, nstart = 25)
k4 <- kmeans(spe.hel, centers = 4, nstart = 25)
k5 <- kmeans(spe.hel, centers = 5, nstart = 25)

# Métodos para determinar o melhor numero de grupos.

fviz_nbclust(spe.hel, kmeans, method = "wss")# metodo do cotovelo
fviz_nbclust(spe.hel, kmeans, method = "silhouette")# metodo silueta

```

Ordenações PCA, NMDS, PCoA & RDA.

```

install.packages(c("BiodiversityR","adespatial"))
library(vegan)
library(BiodiversityR)
library(adespatial)

#### PCoA Analise de Coordenas Principais

# Sempre devemos utilizar uma medida de distancia modo Q.
# Distancia de Hellinger.

bjol <- decostand(spe, method= "hel")
spe.dh <-vegdist(bjol, method='eu') # Distancia de Gellinger

loc.hel <- cmdscale(spe.dh, eig=TRUE) # PCoA com a distancia de Hellinger

#Agora o plot.
plot(loc.hel$points, pch=19, main="Hellinger")
# Ou
biplot(loc.hel)

# Adicionando os Loadings das espécies.
loc.hel<- add.spec.scores(loc.hel, # objeto da PCoA
                          spe.hel, # Matriz das espécies
                          method="pcoa.scores",
                          Rscale=FALSE, scaling=1,
                          multi=1)

# Complementando o grafico.
# Comandos servem somente depois que "chamar" um plot.

```

```

ordispider(loc.hel$points, #pontos do objeto da PCoA
            k2$cluster, # Agrupamento
            label=TRUE,
            col = c("black","red"))
ordihull(loc.hel$points,
          k2$cluster,
          label=TRUE,
          col = c("black","red"))
ordisurf(loc.hel$points,
          env$alt, # variável continua numerica qualquer
          col = 'blue'
          )

##### NMDS #####
*****
# NMDS não retorna a percentagem de cada eixo e sim uma medida de stress.
example_NMDS <- metaMDS(spe.dh ,k=2, trymax=100)
plot(example_NMDS)
NMDS.sco<-scores(example_NMDS)# para conseguir os scores.
stressplot(example_NMDS)

##### RDA #####
*****
# Separamos env em dois tipos de dados
# Physiography (upstream-downstream gradient)
envtopo <- env[, c(1 : 3)]
names(envtopo)
# Water quality
envchem <- env[, c(4 : 10)]
names(envchem)

# Duas formas diferentes de se fazer a RDA:
# Aqui com apenas uma matriz exploratória (env).
spe.rda <- rda(spe.hel ~ ., env)
spe.rda <- rda(spe.hel, env)

# Para termos mais detalhes do resultado.
summary(spe.rda)
R2adj <- RsquareAdj(spe.rda)$adj.r.squared
R2adj # explicação da RDA em %

# Global test of the RDA result
anova(spe.rda, permutations = how(nperm = 999))

# Plot comum: triplot
plot(spe.rda,
      scaling = 1,
      display = c("sp", "lc", "cn"),
      main = "Triplot RDA spe.hel ~ env"
)

## Partial RDA: effect of water chemistry, holding physiography.

```

```

# Agora RDA com duas matrizes exploratórias.

spechem.physio <- rda(spe.hel, envchem, envtopo)
summary(spechem.physio)
plot(spechem.physio)

# Anova para a RDA
anova(spechem.physio, permutations = how(nperm = 999))

##### Partição de Variância #####
*****
# Para visualizar o numero de partições:
showvarparts(2, bg = c("red", "blue"))
showvarparts(3, bg = c("red", "blue", "black"))

# Agora para os dados
spe.part.all <- varpart(spe.hel, envchem, envtopo)
plot(spe.part.all,
     digits = 2,
     bg = c("red", "blue"),
     Xnames = c('envchem', 'envtopo'))

```

Além de mandar por e-mail eu vou anexar todos os arquivos da aula em minha pagina no **GitHub**.