Universidade Federal da Bahia
Instituto Multidisciplinar em Saúde
Curso de extensão: Introdução ao uso do programa R
Prof. Leandro Martins de Freitas e Profª Leila Cruz

**Introdução ao uso do programa R**

# Dia 5: Análise de dados II

Último dia do curso! Hoje vamos ver duas funções para agrupamento, "k means" e "hclust", e teste de comparação de médias/variâncias, a ANCOVA (Análise de Covariância).

As análises de agrupamento são técnicas para simplificação dos dados: o agrupamento simplifica os dados porque reúne observações similares em uma mesma classe. Assim, descrever e analisar um grande número de observações torna-se mais simples porque a tarefa é convertida na análise de menor número de grupos.

O teste ANCOVA é indicado quando se quer levar em consideração a influência de covariáveis na resposta da variável dependente, além do efeito principal da variável independente em questão (Field et al. 2012)[1]. Em geral, os dados consistem em duas variáveis contínuas e uma variável categórica, que divide as variáveis contínuas em dois ou mais grupos (McDonald, 2014)[2].

Vamos aos testes!

## 1 Testes de Agrupamento

Os métodos de agrupamento que vamos trabalhar são chamados de não supervisionados. Esses métodos separam os elementos em grupos sem o conhecimento prévio dos limites de grupos dentro do conjunto de dados.

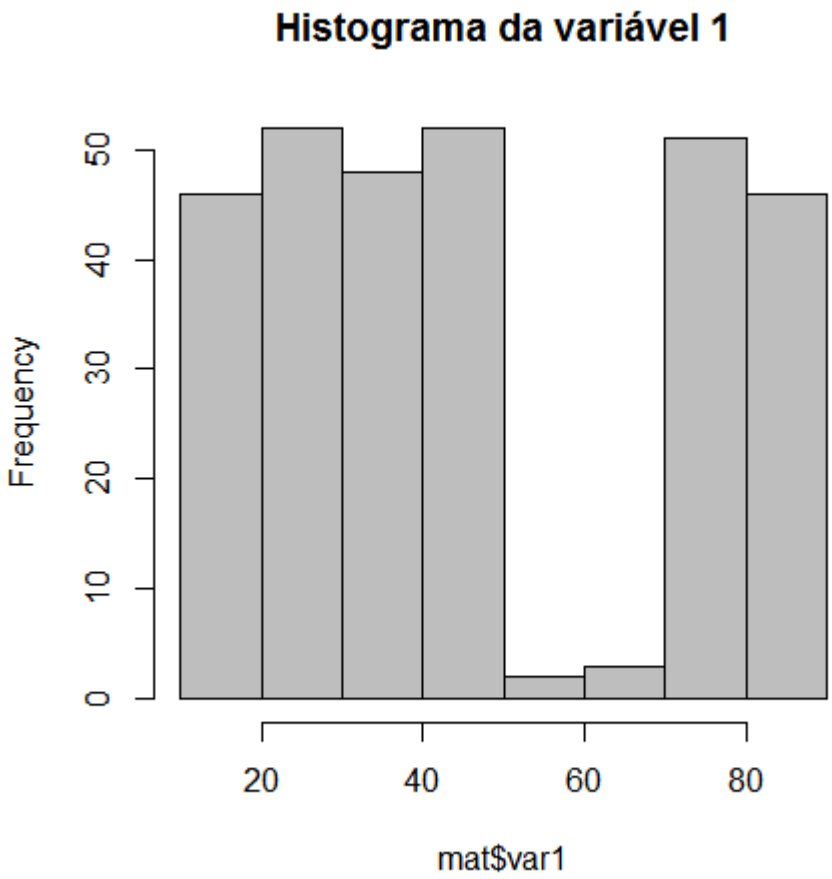
Vamos carregar o primeiro exemplo de dados para clusterização. Esse são dados simulados que apresentam claramente três grupos. Vamos ver como analizar até chegar na definição dos grupos.

```
#carregar os dados exemplo_cluster.csv
>mat <- read.table("mat.csv", row.names=1)
>dim(mat)
```

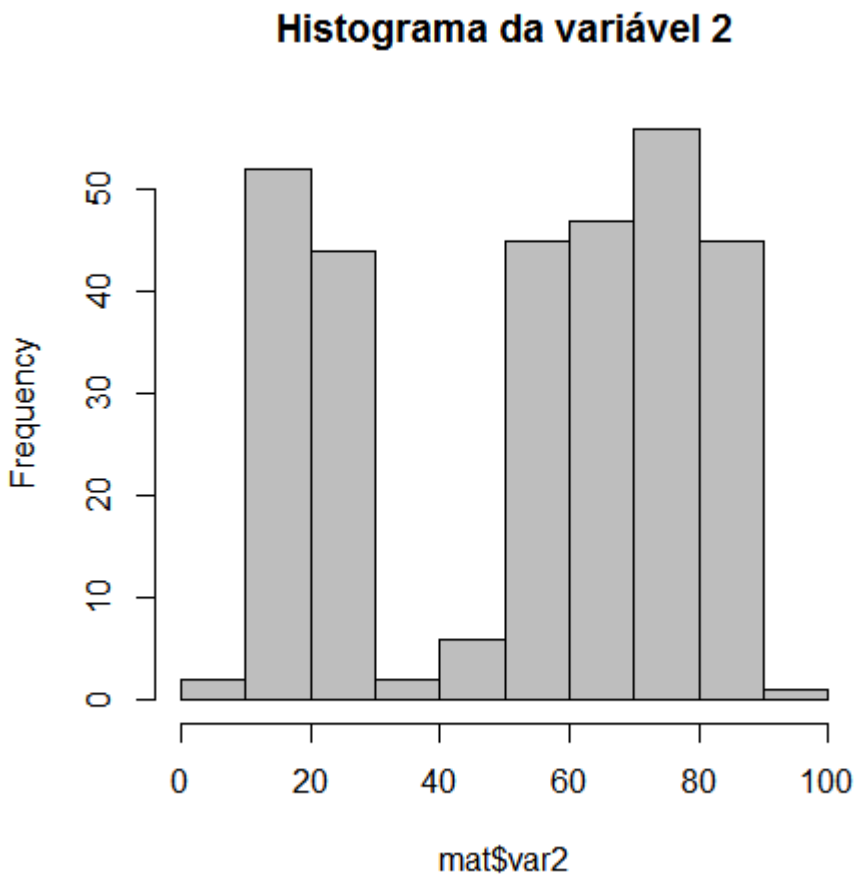
[1] Field, Andy, Jeremy Miles, and Zoë Field. 2012. *Discovering Statistics Using R*. London: SAGE Publications Inc.

[2] McDonald, J.H. 2014. Handbook of Biological Statistics (3rd ed.). Sparky House Publishing, Baltimore, Maryland.

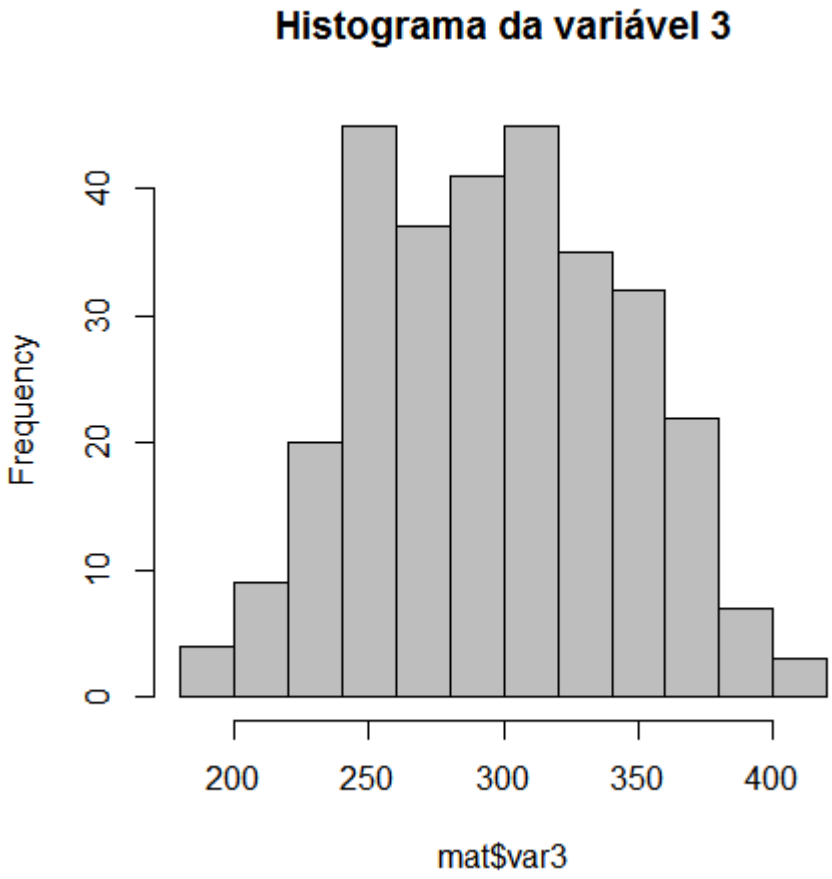
```
>head(mat)

#vamos fazer uma exploração rápida das variáveis usando histograma
#análise exploratória rápida por histograma
hist(mat$var1, breaks="sturges", col="gray",
     main="Histograma da variável 1")
```



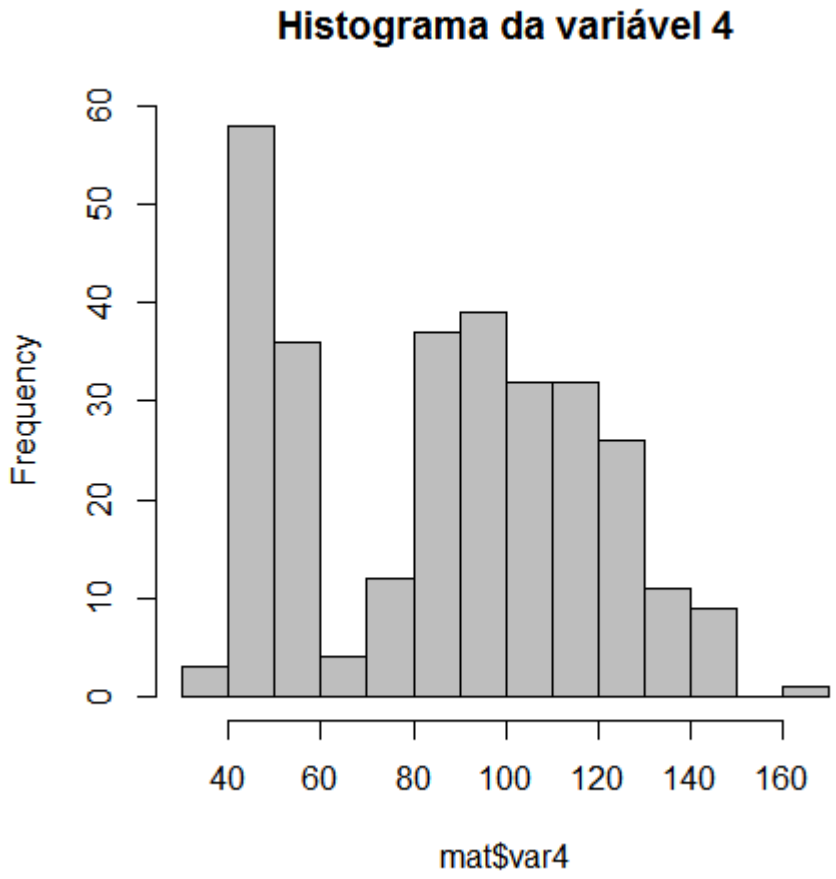

```
hist(mat$var2, breaks="sturges", col="gray",
     main="Histograma da variável 2")
```




```
hist(mat$var3, breaks="sturges", col="gray",
     main="Histograma da variável 3")
```




```
hist(mat$var4, breaks="sturges", col="gray",
     main="Histograma da variável 4")
```




```
#calculando a distancia entre as amostras
#com base nas variáveis
>mat_dist <- dist(mat)
```

Como os estudos coletam informações complexas devemos fazer uma transformação para avaliar quantos grupos existem. Os estudos apresentam diferentes tipos, escalas e diferentes números de variáveis, nós vamos fazer uma transformação dos dados em uma matriz com apenas duas  dimenssões.

```
#realização do MDS
>mat_MDS <- cmdscale(mat_dist,2)
>dim(mat_MDS)
```

As dimenssões do MDS correspondem ao eixo x e y de um gráfico de dispersão. Dessa forma podemos projetar essas dimensões no gráfico e observar a distribução dos pontos. cada ponto corresponde a uma amostra do estudo. Quanto mais próximo dois pontos menor é a diferença entre as váriáveis usada no estudo.

```
>plot(mat_MDS, col ="black",
     xlab="dimensão 1", ylab="dimensão 2", pch=17)
```

Quantos grupos podemos observar nessa projeção? A definição do número de grupos não pode ser abstrata, somente com base na classificação subjetiva de cada pessoa. Entretanto a observação da distribuição pode nos indicar quantos grupos devem existir nos dados.

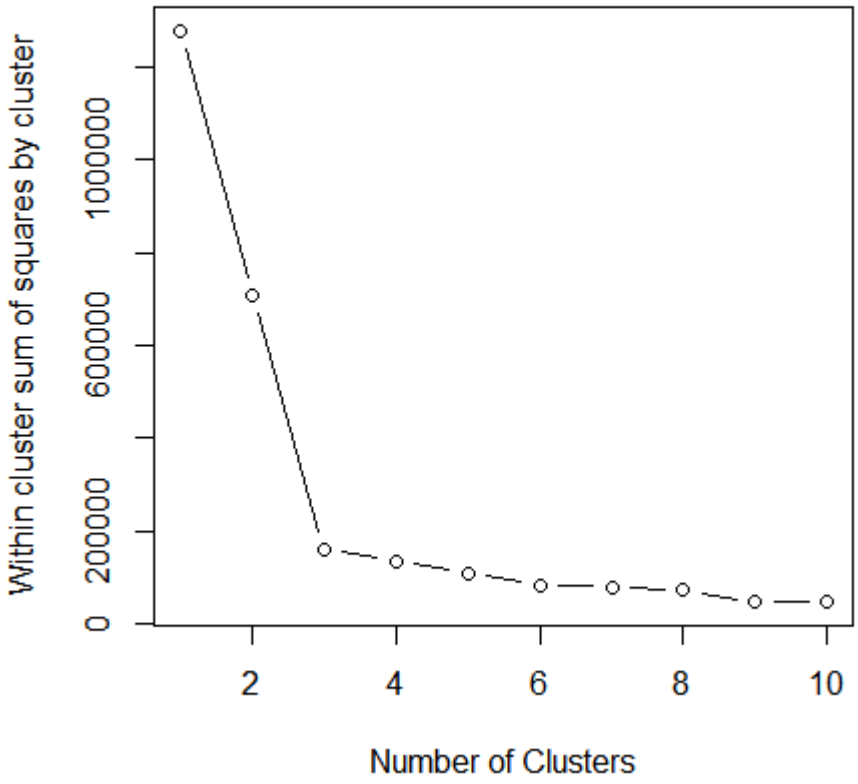
Vamos usar duas formas para estimar o número de grupos.

A primeira forma (Within groups sum of squares) avalia a diversidade dentro dos grupos e fora dos grupos. O método vai dividindo os grupos e a cada divisão que é realizada é mediada a diversidade. Quando a diversidade cai rapidamente com a divisão dos grupos significa que devemos continuar dividindo. Quando a diversidade cai lentamente devemos parar de dividir os grupos.

O R permite que nos criemos nossas próprias funções. Isso cria uma diversidade de análises que podem ser feitas de forma específica para o nosso conjunto de dados.

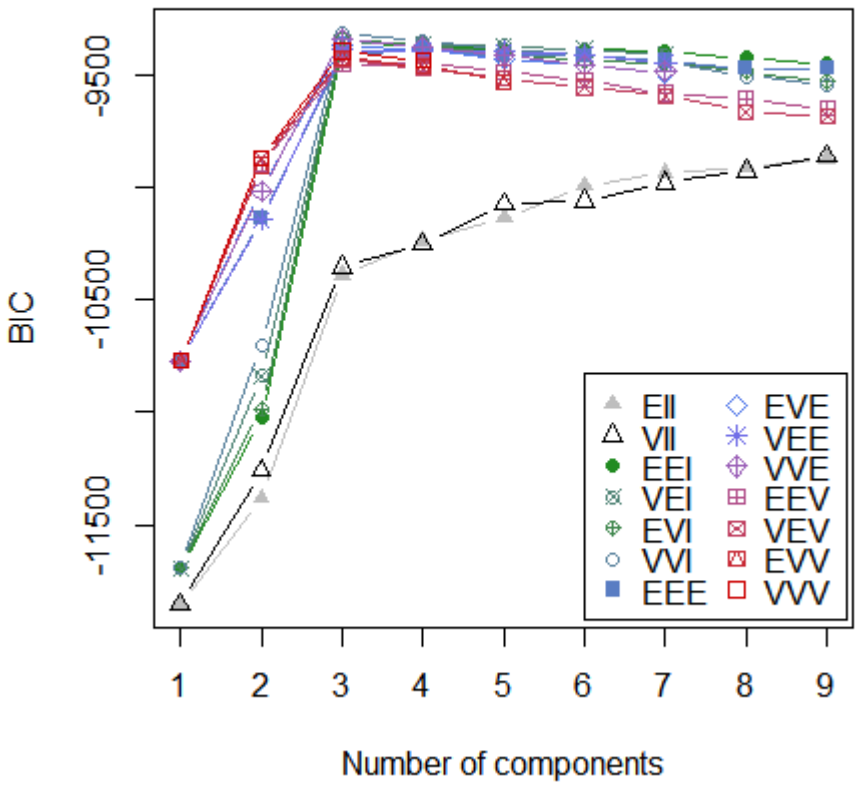
Para avaliar o número de grupos foi criada essa função.

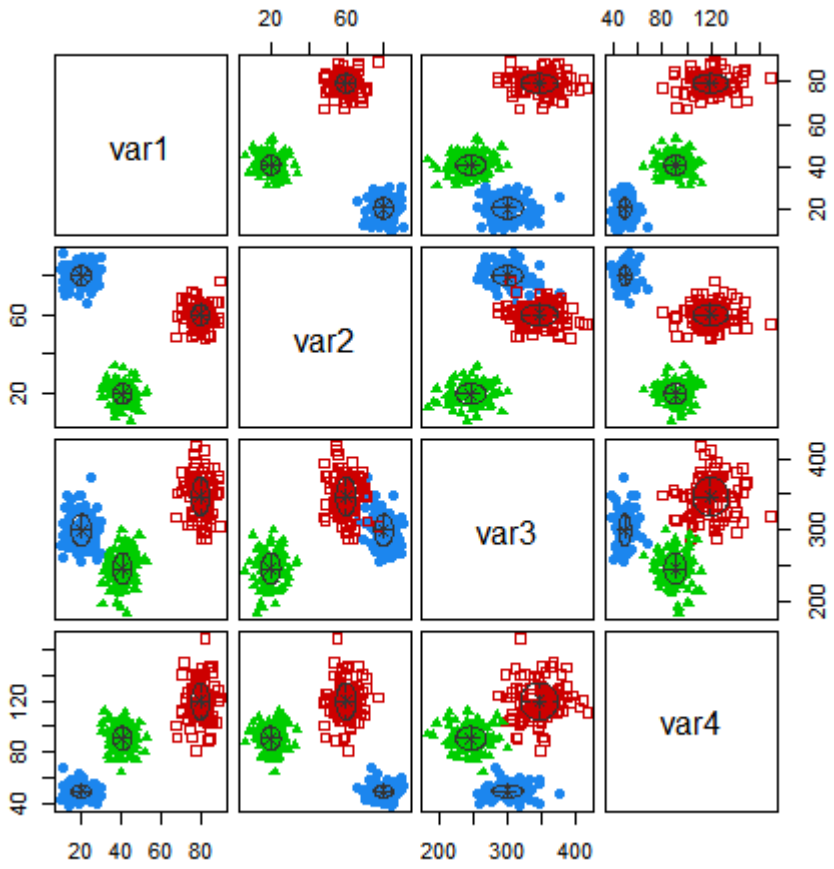
```
#função que avalia o numero de cluster
plotNumCluster <- function (Matrix, maxCluster){
    withinss <- var(Matrix)
    #variation of the whole data
    withinss <- (nrow((Matrix)-1)*sum(withinss[withinss > 0]))
    for (i in 2:maxCluster) {
        withinss[i] <- sum(kmeans(Matrix, i)$withinss)
    }
        plot(1:maxCluster, withinss, type="b", xlab="Number of
Clusters",
             ylab="Within cluster sum of squares by cluster")
}

>plotNumCluster(mat_MDS, 10)
```




A segunda forma usaremos a análise do pacote mclust para estimar o número de grupos. Esse pacote apresenta diversas funções para análise de grupos. Vamos usar a função Mclust() para estimar o número de grupos. Essa função trabalha com diversos métodos para estimar o número de grupos.

```
#avaliação do numero de cluster 2
#instalando a library mclust
#install.packages("mclust", dependencies=T)
>library(mclust)

# Realizando a avaliação do número de grupos - Clustering
>mat_avalia_cluster = Mclust(mat)
#devemos escolher entre dois gráficos.
# o primeiro estima o numero de grupos
# o segundo mostra a classificação dos grupos com base na divisão
que apresenta melhor resultados dos métodos de avaliação.
>summary(mat_avalia_cluster)

>plot(mat_avalia_cluster,  what = c("BIC", "classification"))
```


BIC
-9500
-10500
-11500
1
2
3
4
5
6
7
8
9
Number of components
EII
VII
EEI
VEI
EVI
VVI
EEE
EVE
VEE
VVE
EEV
VEV
EVV
VVV



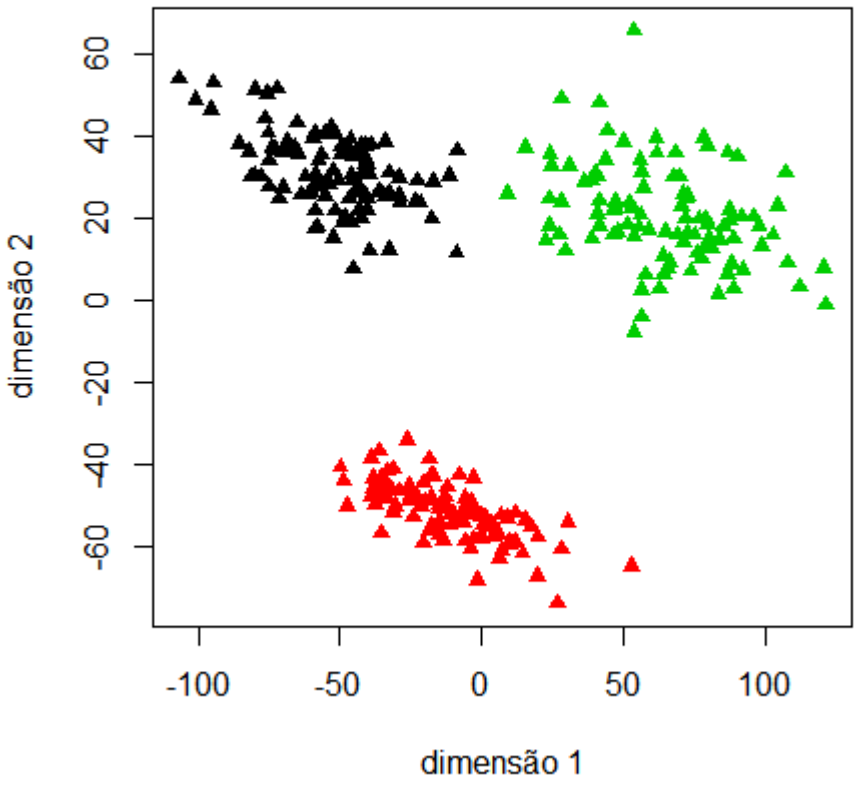

```
#o número de grupos é igual nas duas abordagens?
```

Agora vamos realizar a divisão dos grupos usando a função kmeans(). Essa função simplesmente divide os pontos da projeção no número de grupos escolhido com base na Within groups sum of squares. Nos devemos passar o número de grupos e a função divide da "melhor" forma possivel os elementos nesses grupos. O número mínimo é dois e máximo é o númro de amostras no estudo.

```
#apos definir o numero de grupos
>mat_kmeans <- kmeans(mat_MDS, 3)

#vamos projetar o gráfico de distribuição dos pontos no espaço novamente.
```

Usamos a informação retornada pela função kmeans() para colorir os pontos no gráfico. Cada uma das cores representa um grupo diferente.

```
>plot(mat_MDS, col =mat_kmeans$cluster,
     xlab="dimensão 1", ylab="dimensão 2", pch=17)
```

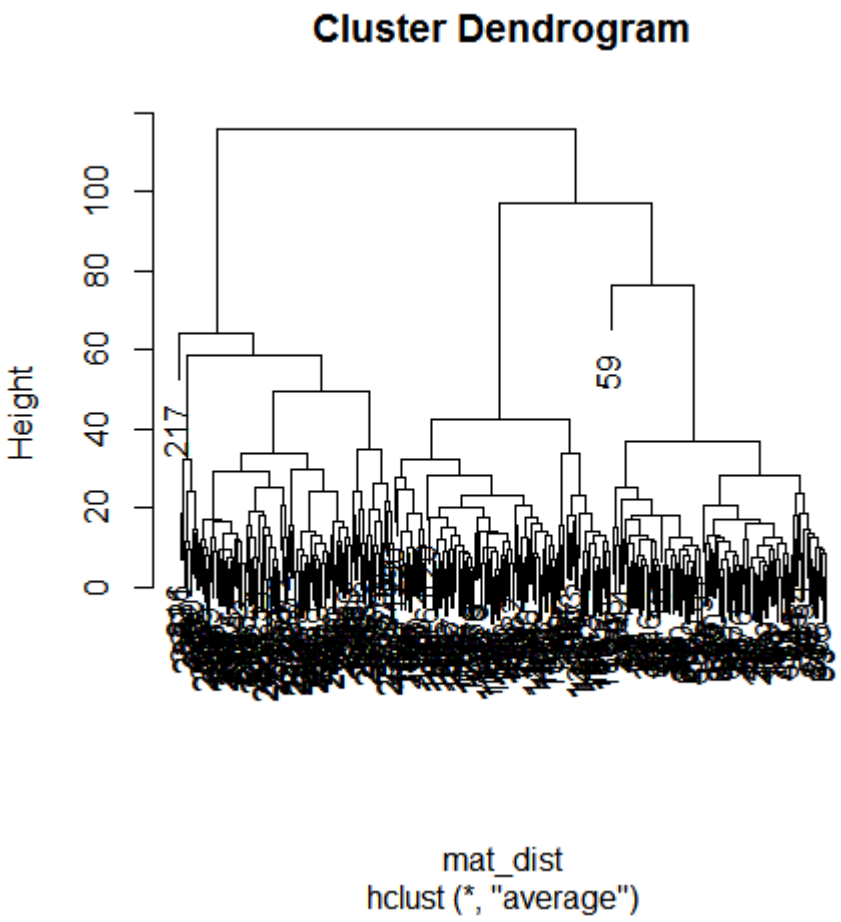



Quantos grupos podemos observar nessa projeção?

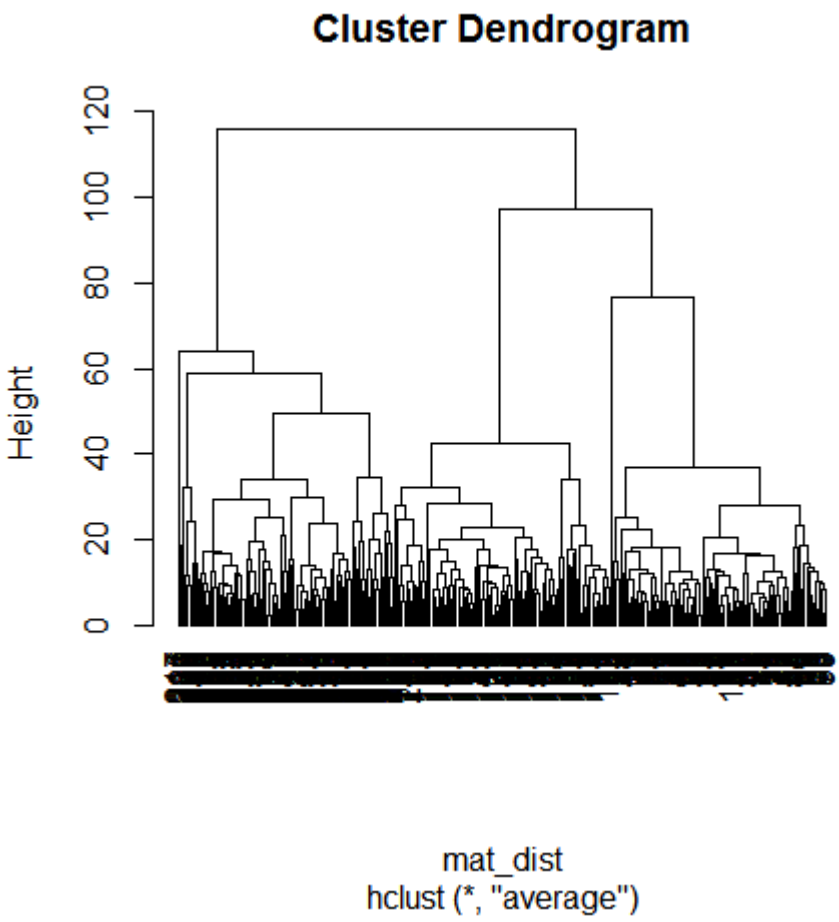
Vamos usar outro método de separação dos grupos. A clusterização hierárquica é a divisão dos grupos com base na similaridade entre os elementos. A distância entre as amostras é usada para conectar em um dendrograma (árvore). Quando as amostras ultrapassam um valor limite é criado um novo ramo. Esse novo ramo é um grupo ou subgrupo dentro da sua amostra.

Vamos usar a função hclust() para fazer essa separação. A função exige uma matriz de distância para calcular o dendrograma. Já fizemos essa matriz antes do MDS.

```
#calculando o dentrograma
>mat_hclusting <- hclust(mat_dist, "ave")
#desenhando o dendrograma
#o nome em cada extremidade é o nome da amostra ( nome das linhas - row names)
>plot(mat_hclusting)
```

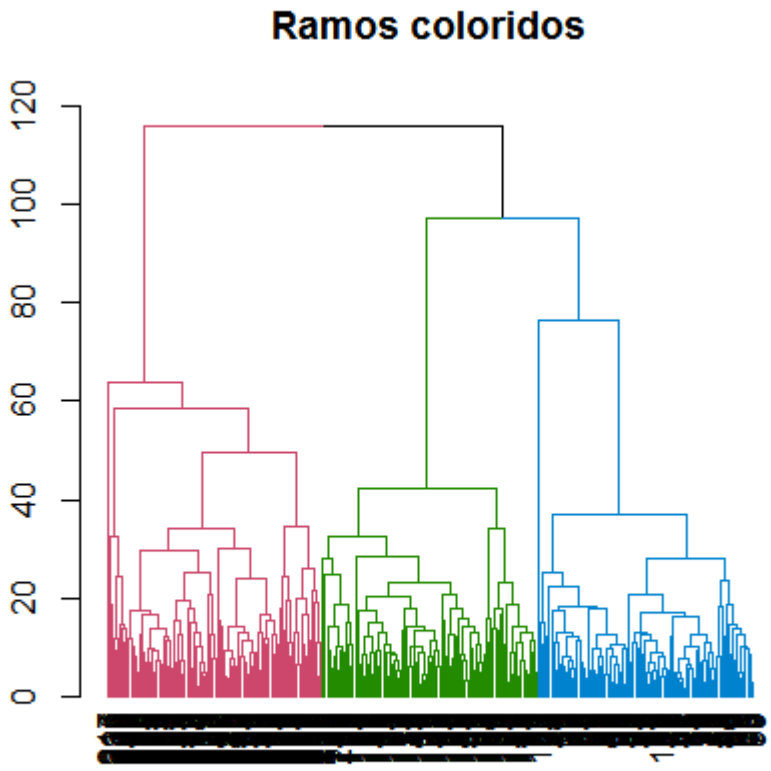



```
#apenas outra forma de apresentar o dendrograma
>plot(mat_hclusting, hang = -1)
```

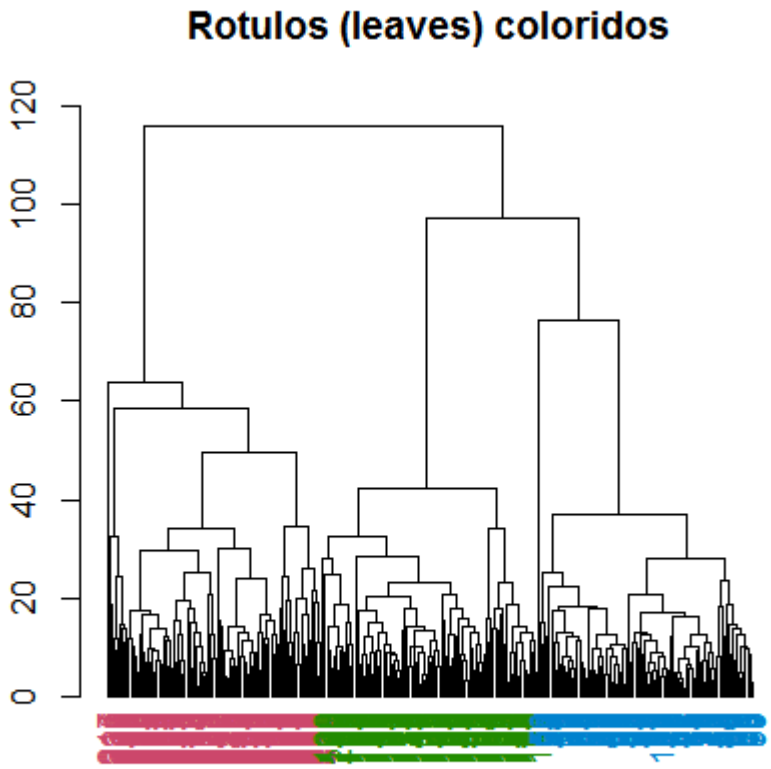



Vamos usar a função cutree() para dividir o dentrograma em grupos. POdemos passar somente um valor numerico que representa o número de grupos, ou passar um vetor de números para fazer a divisão em todos esses valore contidos no vetor.

```
#divisão dos grupos de 1 a 5 grupos
>cutree(mat_hclusting, k = 1:5) #k = 1 is trivial

# install.packages("dendextend")
>library(dendextend)
>dend <- as.dendrogram(mat_hclusting)
#ramos coloridos
>dend1 <- color_branches(dend, k = 3)
#rotulos (leaves) coloridos
>dend2 <- color_labels(dend, k = 3)


>plot(dend1, main = "Ramos coloridos")
```




```
>plot(dend2, main = "Rotulos (leaves) coloridos")
```




Exercício.

Fazer análise de grupos usando o dataset matriz_eco.csv