Cluster Hierarquico

Code

O exemplo foi extraído de: Univesity of Cincinnati (http://uc-r.github.io/hc_clustering)

O agrupamento hierárquico é uma abordagem alternativa para o agrupamento k-means para identificar grupos no conjunto de dados. Esse tipo de agrupamento não exige que pré-especifiquemos o número de clusters a serem gerados como é exigido pela abordagem k-means. Além disso, o agrupamento hierárquico tem uma vantagem adicional sobre o agrupamento de K-means, pois resulta em uma atraente representação baseada em árvores das observações, chamado dendrograma.

Para realizar essa técnica, vamos utilzar os seguintes pacotes:

Hide

```
library(tidyverse)  # Manipulação de dados
library(cluster)    # Algoritmos de cluster
library(factoextra) # Algoritmos de cluster e visualização
library(dendextend) # Comparar Dendrogramas
```

Vamos utilzar uma base de dados disponível no R chamada **USArrest**. Essa base contém estatísticas de prisões a cada 100.000 habitantes por assalto, assassinato e estupro para cada um dos 50 estados dos EUA em 1973. Inclui também a porcentagem da população que vive em áreas urbanas:

Hide

```
df <- USArrests
```

Para remover qualquer valor faltante (missings) em nossos dados, podemos digitar:

Hide

```
df <- na.omit(df)
```

Como não queremos que nosso algoritmo dependa de uma variável de valor arbitrário, vamos padronizar os dados utilizando a função scale

Hide

```
df <- scale(df)
head(df)
```

```
               Murder   Assault   UrbanPop         Rape
Alabama    1.24256408 0.7828393 -0.5209066 -0.003416473
Alaska     0.50786248 1.1068225 -1.2117642  2.484202941
Arizona    0.07163341 1.4788032  0.9989801  1.042878388
Arkansas   0.23234938 0.2308680 -1.0735927 -0.184916602
California 0.27826823 1.2628144  1.7589234  2.067820292
Colorado   0.02571456 0.3988593  0.8608085  1.864967207
```

Cluster hierarquico

Há diferentes tipos de funções disponíveis para criar cluster hierárquicos. Os mais utilizados são:

- hclust e agnes para cluster hierarquico aglomerativo
- diana para cluster divisivo

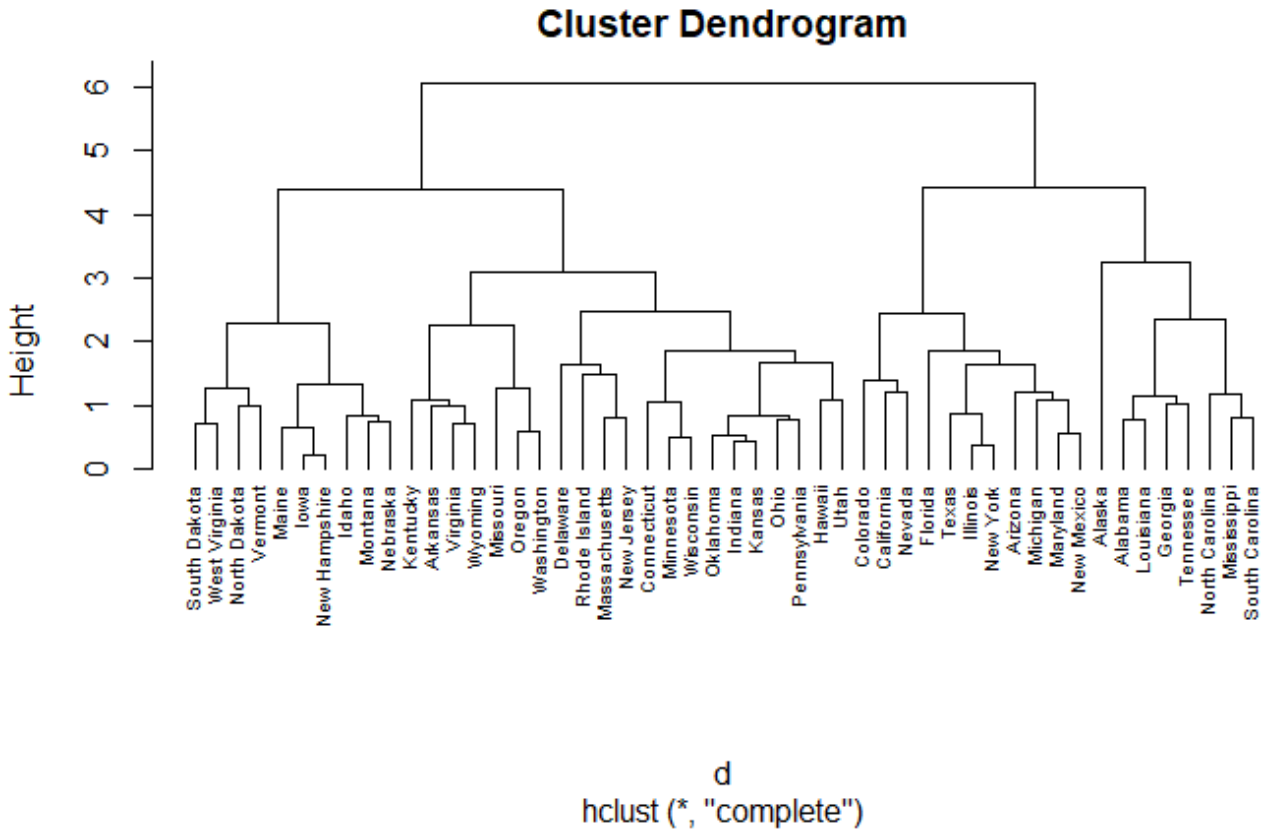
Cluster hierarquico Aglomerativo

Podemos gerar um cluster hierárquico aglomerativo com `hclust`. Primeiro calculamos o valor da dissimilaridade com `dist` e definimos algum método de aglomeração específico (ex. “complete”, “average”, “single”, “ward.D”, etc).

O resultado do dendrograma pode ser visualizado abaixo:

Hide

```
# Matriz de dissimilaridade
d <- dist(df, method = "euclidean")
# Cluster hierarquico utilizando  "Complete Linkage"
hc1 <- hclust(d, method = "complete" )
# Dendrograma
plot(hc1, cex = 0.6, hang = -1)
```



Alternativamente, podemos utilizar a função `agnes`. Essas funções se comportam de forma bem semelhantes, no entanto, com a função `agnes` também podemos obter o coeficiente aglomerativo, que mede a quantidade de estrutura de agrupamento encontrada no cluster (valores próximos de 1 sugerem forte estrutura de agrupamento).

Hide

```
# Cluster utilizando agnes
hc2 <- agnes(df, method = "complete")
# Coeficiente aglomerativo
hc2$ac
```

```
[1] 0.8531583
```

Isso nos permite encontrar certos métodos hierárquicos de agrupamento que podem identificar estruturas de cluster mais fortes. Aqui, vemos que o método de Ward foi identificado como a estrutura de agrupamento mais forte dos quatro métodos avaliados.

Hide

```
# métodos para avaliar
m <- c( "average", "single", "complete", "ward")
names(m) <- c( "average", "single", "complete", "ward")
# função para computar o coeficiente
ac <- function(x) {
  agnes(df, method = x)$ac
}
map_dbl(m, ac)
```

```
  average    single  complete      ward
0.7379371 0.6276128 0.8531583 0.9346210
```

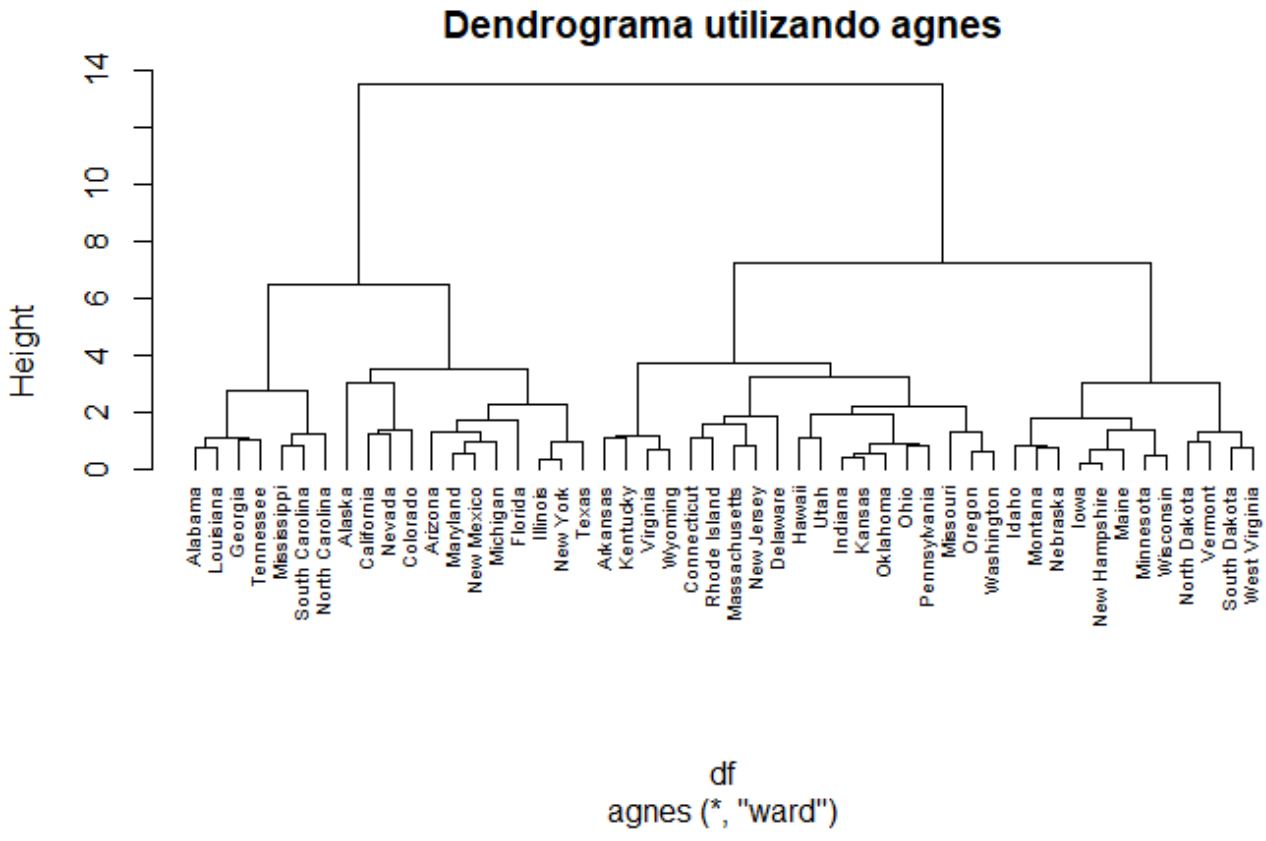
Hide

```
##   average    single  complete      ward
## 0.7379371 0.6276128 0.8531583 0.9346210
```

Da mesma forma que vimos anteriormente, podemos visualizar o dendograma da seguinte forma:

Hide

```
hc3 <- agnes(df, method = "ward")
pltree(hc3, cex = 0.6, hang = -1, main = "Dendrograma utilizando agnes")
```



Cluster Hierárquico Divisivo

A função diana permite criar cluster hierárquico divisivo. O diana funciona de forma parecida com o agnes, mas não precisamos fornecer o método.

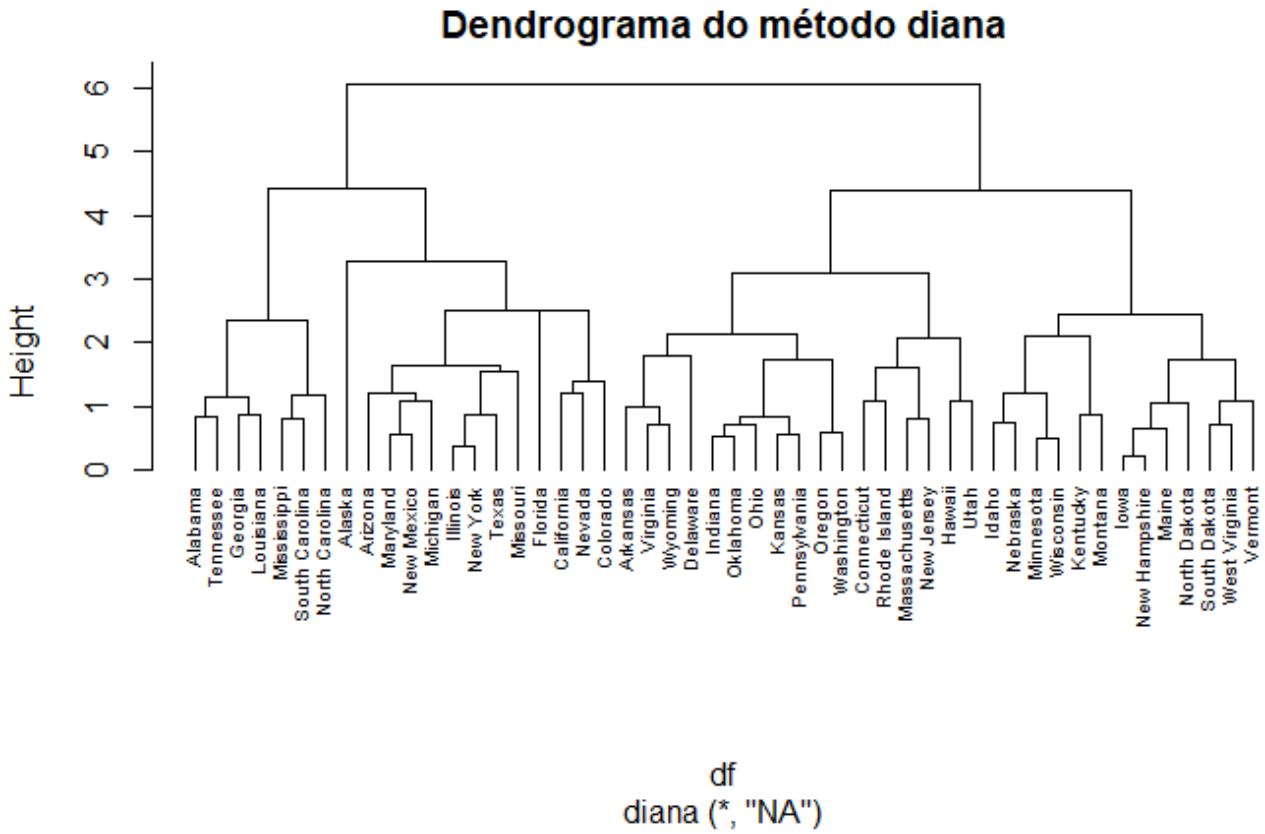
Hide

```
# Cluster hierarquivo divisivo
hc4 <- diana(df)
# Coeficiente divisivo
hc4$dc
```

```
[1] 0.8514345
```

Hide

```
## [1] 0.8514345
# plotar dendrograma
pltree(hc4, cex = 0.6, hang = -1, main = "Dendrograma do método diana")
```



Trabalhando com dendrogramas

Nos dendrogramas anteriores, vimos que cada folha corresponde a uma observação. À medida que avançamos, as medidas que são semelhantes entre si são combinadas em ramos, que são fundidas no próximo nível.

A altura (height) da fusão, fornecida no eixo vertical, indica a (des)similaridade entre duas observações. Quanto maior a altura da fusão, menor será o volume das observações. **Note que, conclusões sobre a proximidade de duas observações podem ser desenhadas apenas com base na altura em que os ramos que contêm essas duas observações são fundidos. Não podemos usar a proximidade de duas observações ao longo do eixo horizontal como um critério de similaridade**.

A altura do corte para o dendrograma controla o número de clusters obtidos. Ele desempenha o mesmo papel que o k no método de k-means. Para identificar subgrupos, podemos cortar o dendrograma utilizando a função cutree:

Hide

```
# Método Ward
hc5 <- hclust(d, method = "ward.D2" )
# Cortar a árvore em 4 clusters (vá em frente, varie o K e veja o que acontece!!)
sub_grp <- cutree(hc5, k = 4)
# Número de observações em cada cluster
table(sub_grp)
```

```
sub_grp
 1  2  3  4
 7 12 19 12
```

Hide

```
## sub_grp
##  1  2  3  4
##  7 12 19 12
```

Além disso, podemos utilizar a saída do cutree para adicionar o número do cluster na qual cada observação pertence aos nossos dados originais.

Hide

```
USArrests %>%
  mutate(cluster = sub_grp) %>%
  head
```

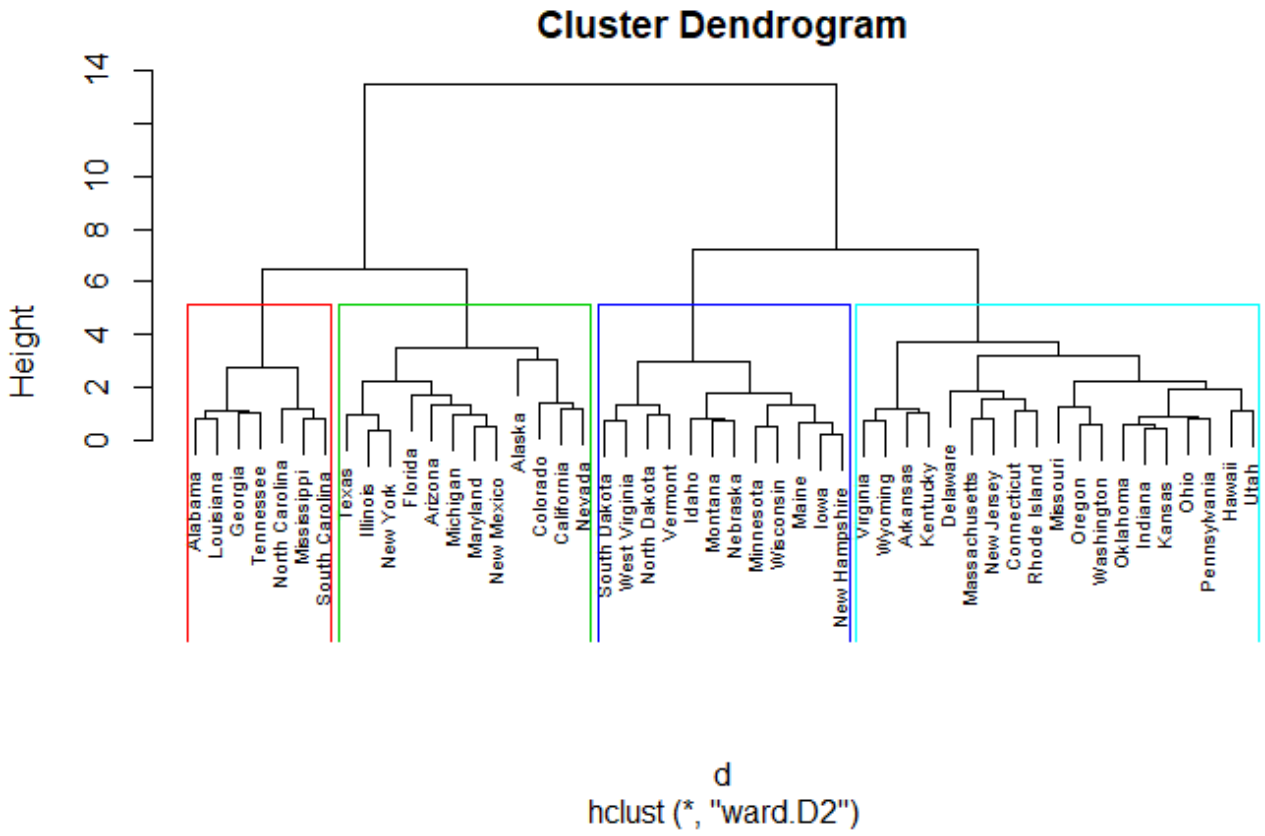
	Murder <dbl>	Assault <int>	UrbanPop <int>	Rape <dbl>	cluster <int>
1	13.2	236	58	21.2	1
2	10.0	263	48	44.5	2
3	8.1	294	80	31.0	2
4	8.8	190	50	19.5	3
5	9.0	276	91	40.6	2
6	7.9	204	78	38.7	2

6 rows

Também é possível destacar cada cluster em nosso dendrograma para deixar o dendrograma amigável e fácil de ser entendido por pessoas “não-técnicas”. Utilizamos o argumento *border* para especificar a cor da borda para os retângulos:

Hide

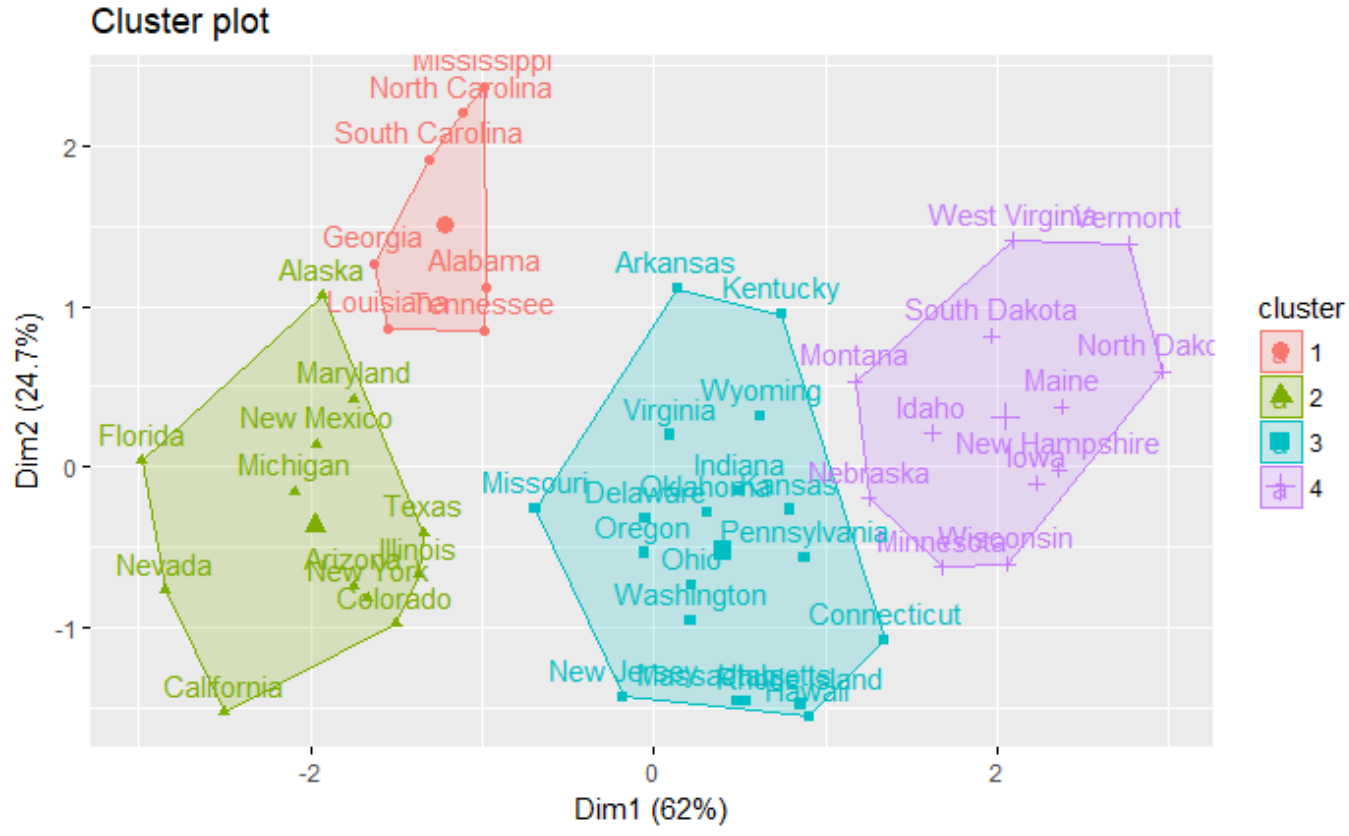
```
plot(hc5, cex = 0.6)
rect.hclust(hc5, k = 4, border = 2:5)
```



Podemos utilizar a função fviz_cluster do pacote factoextra para visualizar o resultado em um *scatter plot* ou *Gráfico de Dispersão* já divido por clustes. Veja que fica mais fácil de visualizar e explicar.

Hide

```
fviz_cluster(list(data = df, cluster = sub_grp))
```



Para podar a árvore para outros métodos utilizando cutree com o agnes e diana podemos rodar da seguinte maneira:

Hide

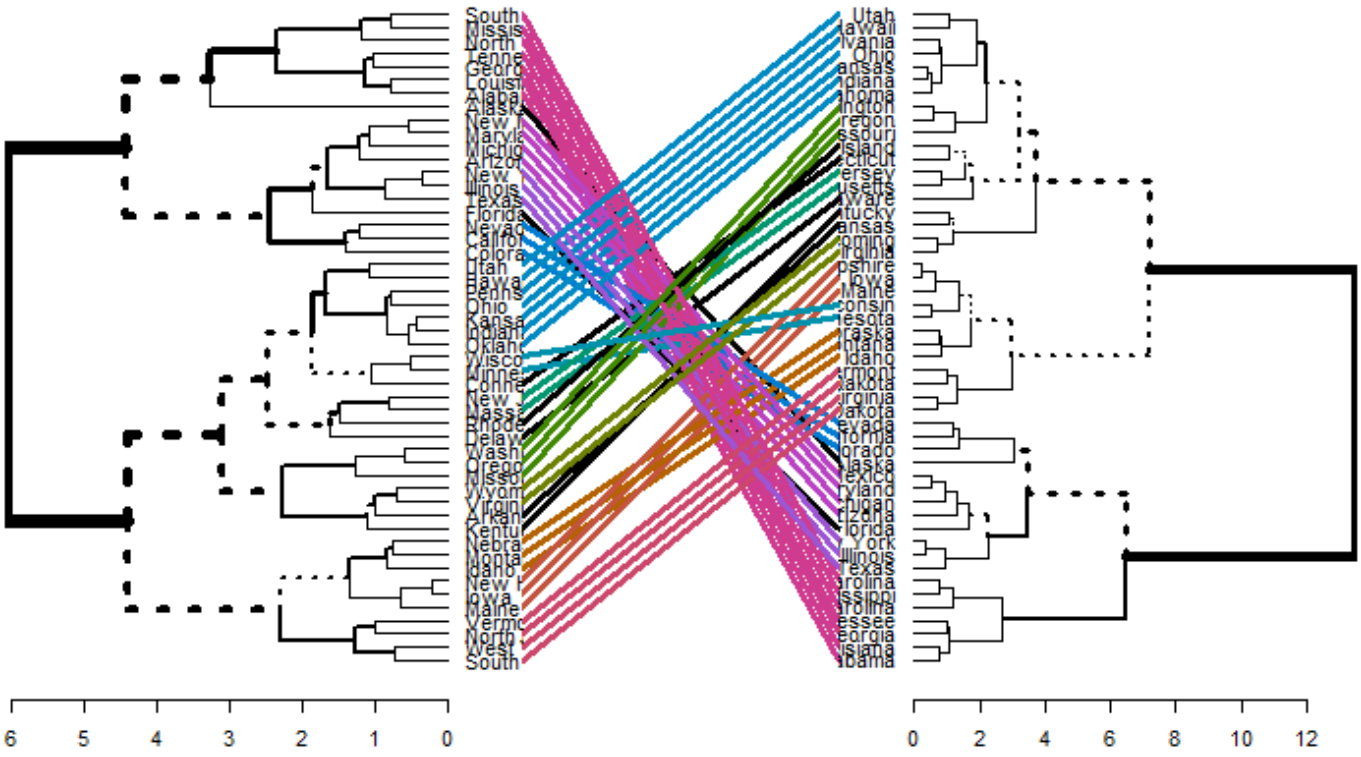
```
# Podar a árvore agnes() em 4 grupos
hc_a <- agnes(df, method = "ward")
cutree(as.hclust(hc_a), k = 4)

# podar a árvore diana() em 4 grupos
hc_d <- diana(df)
cutree(as.hclust(hc_d), k = 4)
```

Por fim, também podemos comparar dois dendrogramas. Vamos comparar o agrupamento hierárquico com ligação completa (complete linkage) versus o método de Ward. A função `tanglegram` plota dois dendrogramas, lado a lado, com os rótulos conectados por linhas.

Hide

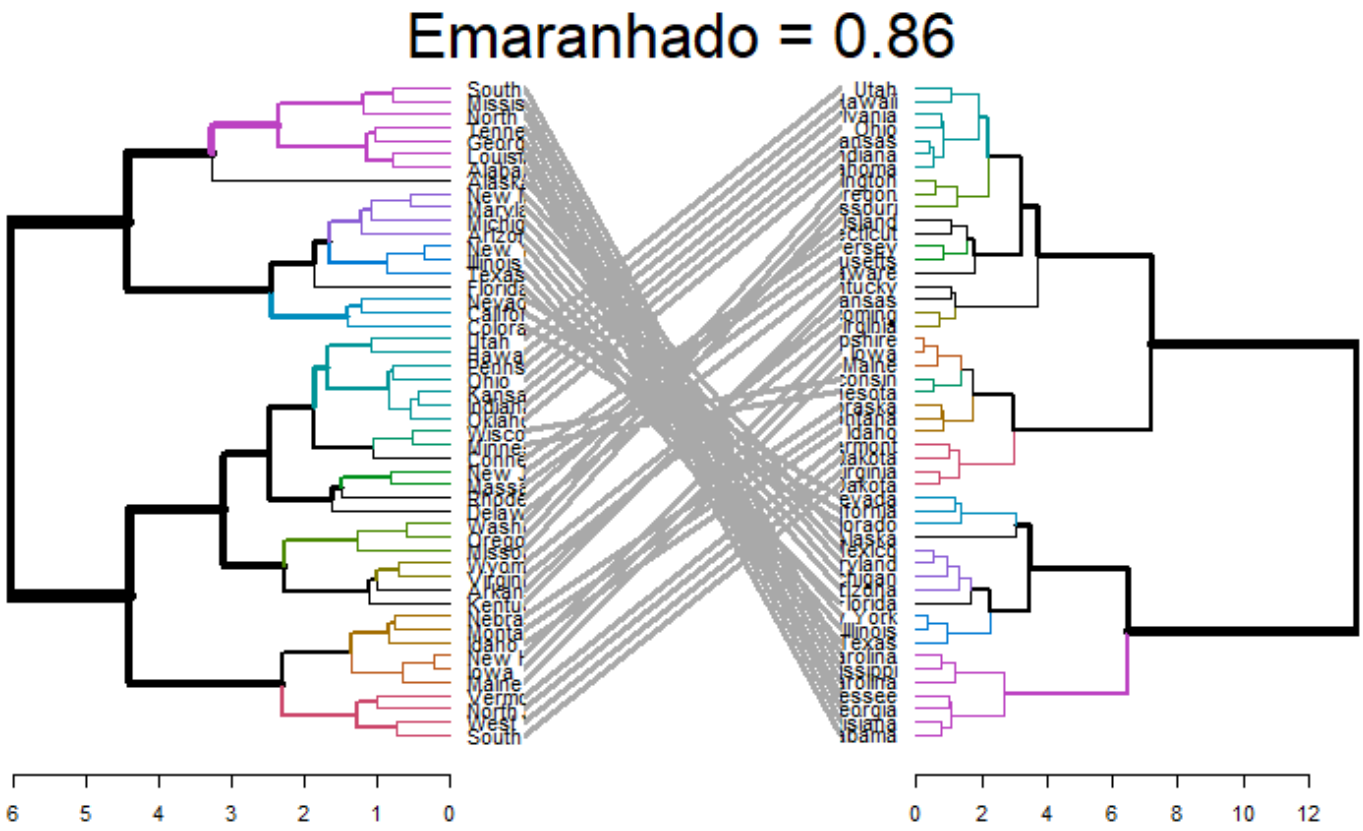
```
# Compute distance matrix
res.dist <- dist(df, method = "euclidean")
# Compute 2 hierarchical clusterings
hc1 <- hclust(res.dist, method = "complete")
hc2 <- hclust(res.dist, method = "ward.D2")
# Create two dendrograms
dend1 <- as.dendrogram (hc1)
dend2 <- as.dendrogram (hc2)
tanglegram(dend1, dend2)
```



A saída exibe nós “únicos”, com uma combinação de rótulos / itens não presentes na outra árvore, destacados com linhas tracejadas. A qualidade do alinhamento das duas árvores pode ser medida usando a função `entanglement` . O emaranhamento é uma medida entre 1 (emaranhamento completo) e 0 (sem emaranhamento). Um coeficiente de emaranhamento inferior corresponde a um bom alinhamento. A saída de `tanglegram` pode ser personalizada usando muitas outras opções da seguinte maneira:

Hide

```
dend_list <- dendlist(dend1, dend2)
tanglegram(dend1, dend2,
  highlight_distinct_edges = FALSE, # Turn-off dashed lines
  common_subtrees_color_lines = FALSE, # Turn-off line colors
  common_subtrees_color_branches = TRUE, # Color common branches
  main = paste("Emaranhado =", round(entanglement(dend_list), 2))
  )
```



Determinando o número ideal de Cluster

Método Elbow

Lembre-se de que, a idéia básica por trás dos métodos de particionamento de cluster, é definir clusters de modo que a variação dentro do cluster total (conhecida como variação total dentro do cluster ou total na soma do conjunto de quadrados) seja minimizada:

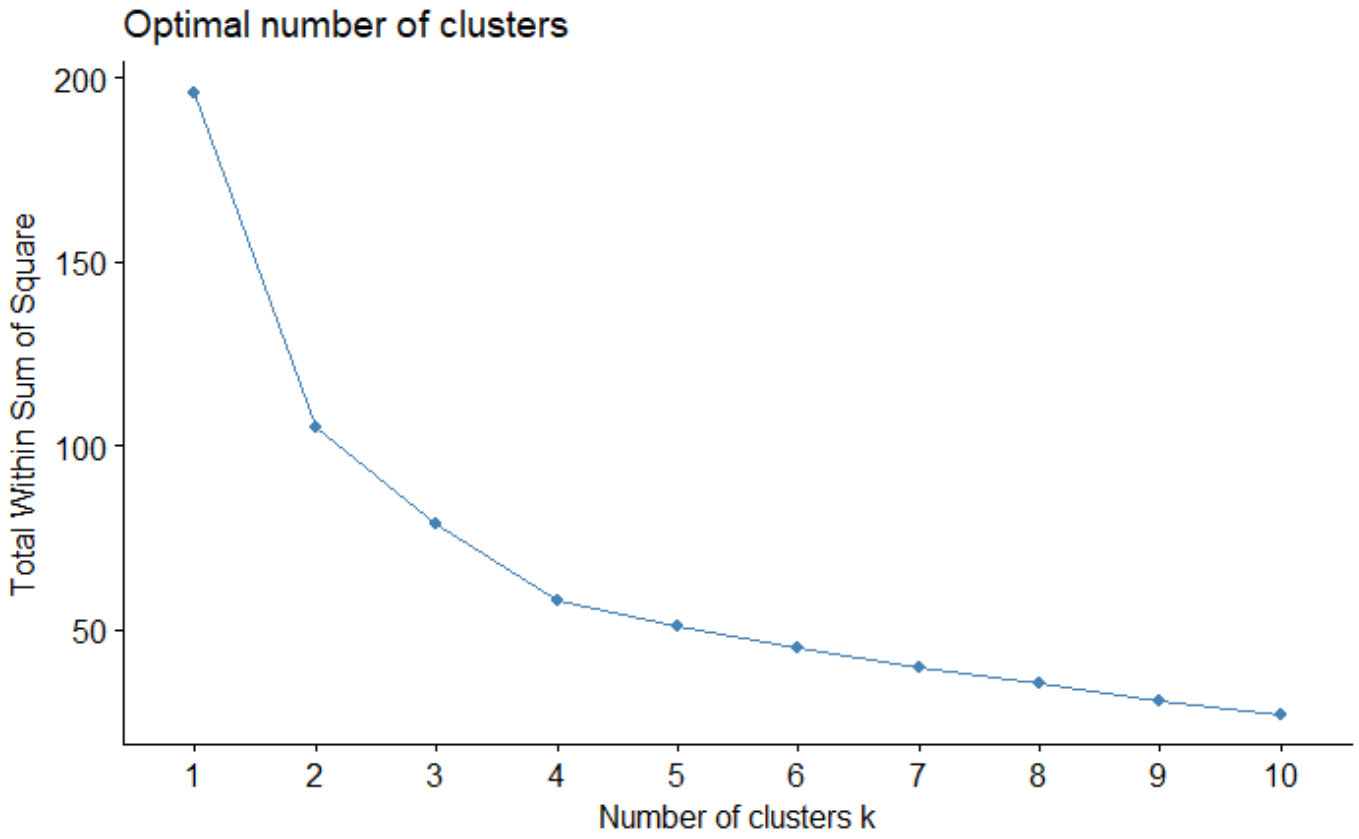
$$minimize(\sum_{i=1}^{n} W(C_k))$$

Onde C_k é o K cluster e $W(C_k)$ é a variação dentro do cluster. Então, o total da soma do quadrado dentro do cluster (wss) mede a compacidade do cluster e queremos que seja tão pequeno quanto possível. Assim, podemos usar o seguinte algoritmo para definir os clusters ótimos:

1. Rodar o algoritmo de agrupamento para diferentes valores de k. Por exemplo, variando k de 1 a 10 clusters
2. Para cada k, calcular a soma total dentro do cluster do quadrado (wss)
3. Traçar a curva de wss de acordo com o número de clusters k.
4. A localização de uma curva (joelho) na trama é geralmente considerada como um indicador do número apropriado de clusters.

Hide

```
fviz_nbclust(df, FUN = hcut, method = "wss")
```



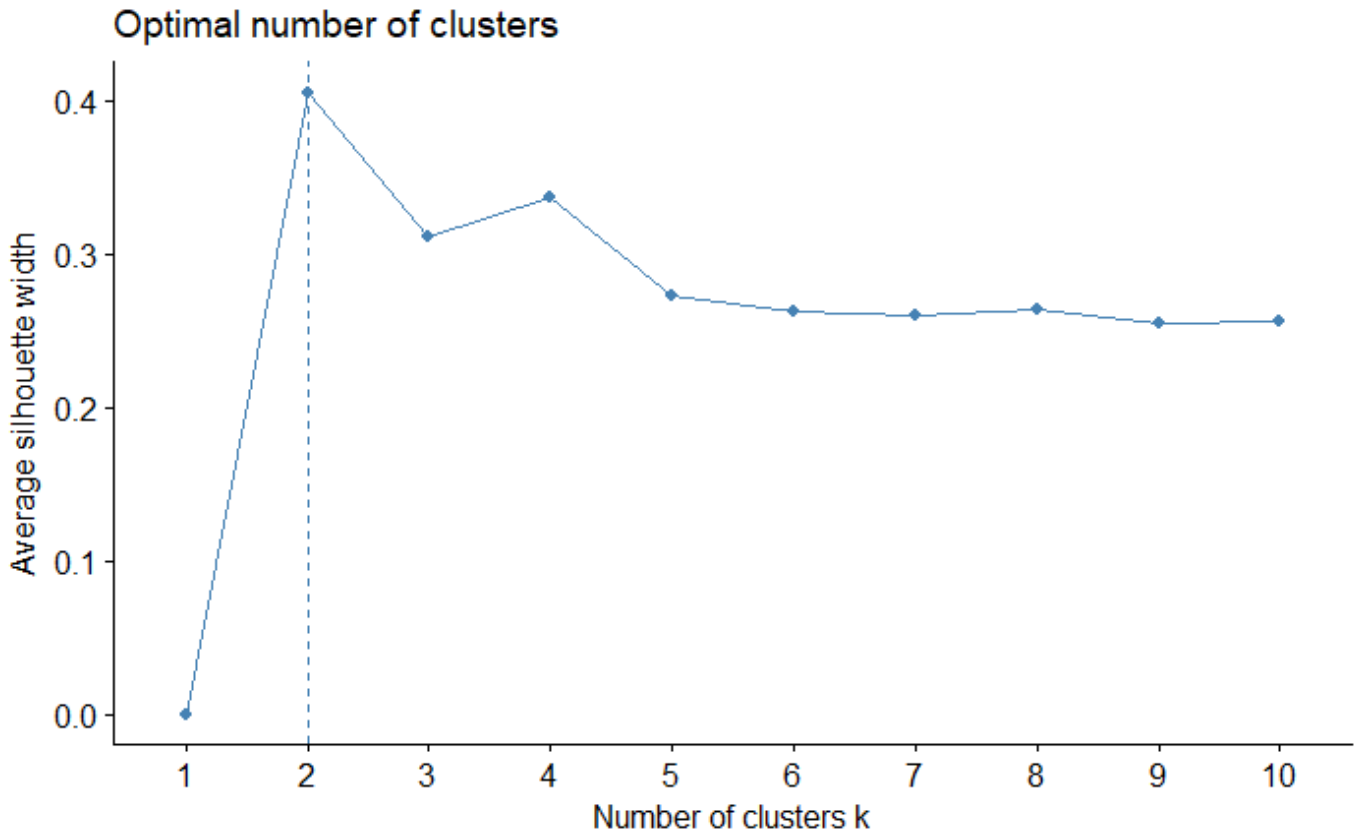
Average Silhoutte Method

A abordagem do método average silhouette mede a qualidade de um cluster. Ou seja, determina o quão bem cada objeto está dentro do seu cluster. Uma largura de silhueta média alta indica um bom agrupamento. O método calcula a silhueta média de observações para diferentes valores de k. O número ótimo de clusters k é aquele que maximiza a silhueta média em uma variedade de valores possíveis para k.

Para rodar o méotodo **average silhoutte** utilizamos o mesmo código do método **Elbow** apenas alterando o parâmentro method

Hide

```
fviz_nbclust(df, FUN = hcut, method = "silhouette")
```



Considerações

Clustering pode ser uma ferramenta muito útil para análise de dados para técnica não supervisionada. No entanto, há uma série de problemas que surgem na realização do agrupamento.

No caso do agrupamento hierárquico, precisamos nos preocupar com alguns pontos:

• Qual a medida de similidade que vamos utilzar? • Qual o tipo de ligação devemos utilzar? • Qual o melhor ponto de corte para cortar o dendrograma?