



Faculdade de Economia, Administração e Contabilidade

Machine Learning

Laboratório de Análise de Dados

Análise de Agrupamento

**Agrupamento K-means
Agrupamento Hierárquico
com a Linguagem R e RStudio**

Prof. Antonio Geraldo da Rocha Vidal

2023

Sumário

Laboratório Agrupamento K-Means	4
Algoritmos de Agrupamento ou Clustering	4
O Melhor Algoritmo de Cluster	5
Preparação dos Dados.....	5
Análise de Agrupamento.....	8
Algoritmo K-means.....	8
Quantidade de Grupos	8
Análise dos Resultados.....	10
Interpretação dos Grupos	14
Conclusão	17
Agora é a sua Vez	17
Evidências da Realização deste Laboratório	17
Referências.....	17
Laboratório Agrupamento Hierárquico.....	18
Operações de pré-processamento para clustering.....	18
Escala.....	18
Imputação de valor ausente.....	19
Algoritmo de clustering hierárquico	19
Dendrograma	21
Medindo a Qualidade dos Clusters	22
Agrupamento Hierárquico em Ação	22
Comparando com o algoritmo de clustering K-means.....	28
Etapa Final: Relatório de Elaboração do Laboratório	28
Conclusão	28
Referências.....	28

Você deve entregar um relatório com os resultados das etapas elaboradas neste laboratório no **e-Disciplinas**, para formatá-lo siga estas orientações:

1. Crie um documento Word e identifique-o com o nome do laboratório, data de elaboração e o seu nome ou do grupo que o elaborou;
2. Crie um tópico para cada resultado que você considerar relevante (manipulação de dados ou resultado de algum processamento) identificando-o com um título e uma breve explicação. Os resultados podem ser imagens de gráficos gerados ou de listas de valores ou dados de resultados

obtidos. Não devem ser incluídos os *scripts* ou instruções de processamento utilizados, inclua apenas os resultados que você considerar relevantes.

3. No final do relatório crie um último tópico denominado “Conclusões” e elabore comentários, sugestões e conclusões sobre o que você pode aprender com a elaboração deste laboratório.
4. O relatório final deve ser entregue no e-Disciplinas no **formato PDF**.

Esta apostila introdutória pode conter erros, falhas ou imprecisões. Se você identificar algum problema por favor informe através do e-mail vidal@usp.br para que a correção possa ser providenciada.

Esta apostila não é autoral, tem objetivo estritamente didático como material de apoio para a disciplina. Foi desenvolvida através da compilação dos diversos textos e materiais citados na bibliografia.



Obrigado!

Laboratório Agrupamento K-Means

A técnica de agrupamento ou clustering é uma técnica de aprendizado não supervisionado. É a tarefa de agrupar um conjunto de objetos de uma forma que os objetos no mesmo grupo ou cluster são mais similares do que os objetos em outros grupos ou clusters diferentes. Similaridade é um valor que reflete a força de relacionamento entre objetos descritos pelos dados. A técnica de clustering é usada principalmente para mineração exploratória de dados. Ela é aplicada em muitos campos, como aprendizado de máquina, reconhecimento de padrões, análise de imagem, recuperação de informações, bioinformática, compactação de dados e computação gráfica.

Neste primeiro laboratório introdutório, você:

- Estudará diferentes métodos de clustering, como conectividade, centroide, distribuição e densidade.
- Estudará o algoritmo k-means clustering, um exemplo de agrupamento baseado em distância a um centroide.
- Descobrirá como obter dados de indicadores macroeconômicos de países utilizando a API (interface) do Banco Mundial (World Bank);
- Analisará grupos de países de acordo com esses indicadores, usando o algoritmo de agrupamento **K-means**;
- Descobrirá em qual grupo o Brasil se encontrava em 2014 e quais são os outros países deste mesmo grupo.
- Descobrirá em qual grupo o Brasil se encontrava em 2018 e quais são os outros países deste mesmo grupo.

Algoritmos de Agrupamento ou Clustering

Um agrupamento é uma coleção de objetos de dados onde os objetos de um mesmo grupo são similares entre si e diferentes dos objetos dos demais grupos. A análise de agrupamento corresponde à uma classificação não supervisionada, ou seja, sem classes ou grupos pré-definidos. Portanto, você precisará descobrir qual é o número de grupos que melhor agrupa os objetos analisados e com isso aumentará o seu conhecimento sobre como estes objetos de distribuem. Como cada grupo identificado terá características semelhantes você poderá utilizar este conhecimento para diversas aplicações, entre elas:

- **Marketing:** descobrir quantos grupos distintos de clientes existem e utilizar esse conhecimento para aplicar diferentes estratégias de relacionamento com cada grupo;
- **Uso do Solo:** identificar áreas com uso semelhante do solo a partir de uma base de dados georreferenciados e aplicar diferentes regras de ocupação do solo visando a preservação ambiental;
- **Seguros:** identificar grupos distintos de clientes e bens segurados para determinação de níveis de preços em apólices;
- **Planejamento Urbano:** identificar grupos de edificações de acordo com seu tipo, dimensões, valor e localização geográfica para aplicar diferentes alíquotas de IPTU;
- **Programas Sociais:** identificar grupos distintos de cidadãos de acordo com suas características, necessidades e localização para aplicar programas sociais adequados.

Um bom método de agrupamento produzirá grupos com alta similaridade intra-grupo, isto é, os objetos de um mesmo grupo são bastante semelhantes, e baixa similaridade inter-grupo, isto é, os objetos de grupos diferentes são bastante diferentes. Entretanto, uma definição precisa para a qualidade do agrupamento é muito difícil e subjetiva, pois depende do problema em análise e dos dados disponíveis para descrever os objetos. Em função disso, resultado sempre dependerá da interpretação do analista.

Algoritmos de clustering podem ser categorizados com base em seu modelo de cluster, que define como eles formam clusters ou grupos. Destacaremos apenas alguns dos principais algoritmos de clustering:

1. Clustering baseado em conectividade: a principal ideia por trás deste algoritmo de agrupamento é que os pontos de dados que estão mais próximos no espaço de dados estão mais relacionados (são mais semelhantes) do que os pontos de dados mais distantes. Os clusters são formados conectando pontos de dados de acordo com a distância. Em distâncias diferentes, os clusters diferentes formarão e podem ser representados usando um diagrama denominado

dendrograma, que mostra porque são chamados também de "aglomeração hierárquica". Esses métodos não produzem um particionamento exclusivo do conjunto de dados, em vez disso, produzem uma hierarquia da qual o usuário ainda precisa selecionar clusters apropriados escolhendo o nível em que deseja agrupar. Eles também não são muito robustos para outliers, que podem aparecer como clusters adicionais ou até mesmo causar outros clusters para mesclar.

2. Agrupamento baseado em centroide: neste tipo de agrupamento, os clusters são representados por um vetor central ou um centroide. Este centroide pode não ser necessariamente um membro do conjunto de dados. Este é um algoritmo iterativo de clustering em que a noção de similaridade é derivada por como achar um ponto de dados para o centroide do cluster. **K-means** é um agrupamento baseado em centroide, e você verá este tópico mais detalhadamente mais adiante.
3. Clustering baseado em distribuição: esse agrupamento está intimamente relacionado às técnicas estatísticas de modelagem distribucional. O agrupamento baseia-se na noção de quão provável é que um ponto de dados pertença a uma determinada distribuição, como a distribuição gaussiana, por exemplo. Os pontos de dados em um cluster pertencem à mesma distribuição. Estes modelos têm uma forte base teórica, no entanto, muitas vezes sofrem de super ajuste (*overfitting*). Modelos de mistura gaussiana, usando o algoritmo de maximização de expectativa são métodos de clustering de distribuição famosos.
4. Métodos baseados em densidade: pesquisam o espaço de dados para áreas de densidade variada de pontos de dados. Os clusters são definidos como áreas de maior densidade dentro do espaço de dados em comparação com outras regiões. Os pontos de dados nas áreas esparsas são geralmente considerados como pontos de ruído e/ou de borda. A desvantagem com esses métodos é que eles esperam algum tipo de guia de densidade ou parâmetros para detectar bordas de cluster. DBSCAN e Óptica são alguns exemplos de algoritmos baseados em densidade.

O Melhor Algoritmo de Cluster

Agora que você já conheceu alguns tipos de algoritmos de clustering, a grande questão é: como você pode identificar o melhor algoritmo para usar? Infelizmente não há um algoritmo melhor do que o outro. Clustering está na percepção do analista! Conforme comentamos, a análise de agrupamento ou clustering é uma tarefa subjetiva e pode haver mais de um algoritmo de clustering correto. Cada algoritmo segue um conjunto diferente de regras para definir a "similaridade" entre os objetos de dados. O algoritmo de clustering mais apropriado para um problema específico geralmente precisa ser escolhido experimentalmente, a menos que haja uma razão matemática para preferir um algoritmo de clustering em relação a outro. Um algoritmo pode funcionar bem em um determinado conjunto de dados, mas falhar para um tipo diferente de dados.

Preparação dos Dados

Nesta seção, você trabalhará com o conjunto de dados do Banco Mundial (World Bank), que contém informações sobre indicadores econômicos e sociais de todos os países do mundo. Estes indicadores podem ser consultados no seguinte endereço do Banco Mundial:

<https://databank.worldbank.org/reports.aspx?source=world-development-indicators>

Note que pode ocorrer alguma dificuldade na obtenção dos dados, uma vez que o Banco Mundial periodicamente atualiza o seu site. Entre em contato com o professor caso não consiga obtê-los.

Inicialmente vamos instalar os pacotes R que precisamos para importar e visualizar os dados que utilizaremos nesta análise:

```
install.packages("WDI")
library(WDI) # baixar os dados do World Bank
library(magrittr)
install.packages("formattable")
library(formattable)
```

O processo de importação dos dados do World Bank é feito de maneira automatizada pelo pacote WDI usando a função `WDI()`. Como é necessário inserir o código do indicador desejado, no exemplo abaixo utilizamos a função `WDIsearch()` para buscar o código do indicador relacionado a, por exemplo, inflação:

```
WDIsearch("Inflation")
##      indicator      name
[1,] "FP.WPI.TOTL.ZG"  "Inflation, wholesale prices (annual %)"
[2,] "FP.FPI.TOTL.ZG"  "Inflation, food prices (annual %)"
[3,] "FP.CPI.TOTL.ZG"  "Inflation, consumer prices (annual %)"
[4,] "NY.GDP.DEFL.KD.ZG.AD" "Inflation, GDP deflator: linked series (annual %)"
[5,] "NY.GDP.DEFL.KD.ZG"  "Inflation, GDP deflator (annual %)"
[6,] "NY.GDP.DEFL.87.ZG"  "Inflation, GDP deflator (annual %)"
```

Verifique que há vários, mas vamos utilizar o indicador geral de inflação, cujo código é “FP.CPI.TOTL.ZG”. Repetimos o mesmo procedimento para outros indicadores que podemos escolher para esta análise: Inflação Anual (%), PIB per capita (USD), Crescimento do Anual do PIB (%) e Taxa de Desemprego (%).

```
# lista de indicadores econômicos dos países:
lista_indicadores <- c("FP.CPI.TOTL.ZG", # inflação (%)
                      "NY.GDP.PCAP.CD", # Pib per capita (USD)
                      "NY.GDP.MKTP.KD.ZG", # crescimento do PIB anual (%),
                      "SL.UEM.TOTL.ZS" # Desemprego (%)
)
```

Usaremos inicialmente o ano de 2014 como ano de referência, depois você utilizará o ano de 2017 para fazer uma comparação da situação do Brasil nestes dois anos.

```
df2014 <- WDI(indicator = lista_indicadores, country = "all", start = 2014, end = 2014,
extra = TRUE)
str(df2014)
# Resultado dos Dados Carregados
'data.frame':      264 obs. of  14 variables:
 $ iso2c      : chr  "1A" "1W" "4E" "7E" ...
 $ country     : chr  "Arab World" "World" "East Asia & Pacific (excluding high
income)" "Europe & Central Asia (excluding high income)" ...
 $ year       : int   2014 2014 2014 2014 2014 2014 2014 2014 2014 2014 ...
 $ FP.CPI.TOTL.ZG : num   2.77 2.29 3.14 2.53 6.67 ...
 .. attr(*, "label")= chr "Inflation, consumer prices (annual %)"
 $ NY.GDP.PCAP.CD : num   7498 10929 6283 9886 1494 ...
 .. attr(*, "label")= chr "GDP per capita (current US$)"
 $ NY.GDP.MKTP.KD.ZG: num    2.46 2.84 6.76 2.23 6.99 ...
 .. attr(*, "label")= chr "GDP growth (annual %)"
 $ SL.UEM.TOTL.ZS  : num   10.21 5.19 4.03 7.42 2.8 ...
 .. attr(*, "label")= chr "Unemployment, total (% of total labor force) (modeled ILO
estimate)"
 $ iso3c       : Factor w/ 304 levels "ABW","AFG","AFR",...: 8 297 85 88 239 6 9 2
13 5 ...
 $ region      : Factor w/ 8 levels "Aggregates","East Asia & Pacific",...: 1 1 1 1
1 3 5 7 4 3 ...
 $ capital     : Factor w/ 212 levels "", "Abu Dhabi",...: 1 1 1 1 1 10 2 80 167 191
 $ longitude   : Factor w/ 212 levels "", "-0.126236",...: 1 1 1 1 1 70 187 195 37
126 ...
 $ latitude    : Factor w/ 212 levels "", "-0.229498",...: 1 1 1 1 1 151 100 122 78
146 ...
 $ income      : Factor w/ 5 levels "Aggregates","High income",...: 1 1 1 1 1 2 2 3
2 5 ...
 $ lending     : Factor w/ 5 levels "Aggregates","Blend",...: 1 1 1 1 1 5 5 4 3 3
# Ou você pode consultar os dados carregados com o comando abaixo
View(df2014)
```

O resultado obtido acima mostra que o data frame não contém dados apenas de países, mas também de unidades agregadas, como o mundo, o mundo árabe, a América Latina etc. Por isso, vamos remover as unidades agregadas:

```
df2014$region %<>% as.character
# Remover agregados
df2014 <- subset(df2014, region != "Aggregates")
```

A seguir, criamos um conjunto de dados (dataframe) denominado dfi2014 apenas com as variáveis de interesse para a nossa análise:

```
dfi2014 <- df2014[, lista_indicadores]
row.names(dfi2014) <- df2014$country
colnames(dfi2014) <- c("Inflacao", "PIB_per_Capita", "Crescimento_PIB", "Desemprego")
summary(dfi2014)
```

Inflacao	PIB_per_Capita	Crescimento_PIB	Desemprego
Min. : -1.5092	Min. : 248.8	Min. : -25.907	Min. : 0.190
1st Qu.: 0.7263	1st Qu.: 2129.9	1st Qu.: 1.364	1st Qu.: 3.740
Median : 2.3463	Median : 6682.1	Median : 3.122	Median : 6.479
Mean : 3.9374	Mean : 17583.5	Mean : 3.234	Mean : 7.954
3rd Qu.: 5.2221	3rd Qu.: 20249.9	3rd Qu.: 5.037	3rd Qu.: 10.698
Max. : 62.1686	Max. : 189170.9	Max. : 36.524	Max. : 27.517
NA's : 34	NA's : 11	NA's : 12	NA's : 30


```
# Podemos visualizar também estes dados utilizando o comando
View(dfi2014)
```

Duas observações importantes sobre o resultado anterior:

- Para facilitar a interpretação dos resultados da análise, transformaremos a taxa de desemprego em taxa de emprego, pois assim teremos três indicadores que quanto maior forem seus valores, mais pujante é a Economia de seus países;
- Alguns países não contêm dados para alguns dos indicadores. Não há informação, por exemplo, sobre desemprego para 38 países.

Para resolver o problema dos valores ausentes (os NA ou *Not Available*), poderíamos aplicar uma técnica robusta ou realizar uma nova pesquisa de dados, mas como este é um exemplo de análise introdutório, optaremos por simplesmente remover os países que tenham algum dado faltando do nosso conjunto de dados para análise.

```
dfi2014 <- na.omit(dfi2014)
dfi2014$Desemprego <- 100 - dfi2014$Desemprego
names(dfi2014)[4] <- "Emprego"
View(dfi2014)
```

O comando View(dfi2014) é equivalente a verificar na janela *Environment* do RStudio o dataframe dfi2014 clicando o botão à direita  .

	Inflacao	PIB_per_Capita	Crescimento_PIB	Emprego
United Arab Emirates	2.34626866	43751.8389	4.398696682	97.939
Afghanistan	4.67399604	613.8563	2.724543365	98.265
Albania	1.61304235	4578.6679	1.770000297	82.510
Armenia	2.98130869	3986.2316	3.600000001	82.502
Angola	7.28038730	5408.4105	4.822625549	92.571
Austria	1.60580415	51717.4959	0.661274653	94.380
Australia	2.48792271	62510.7912	2.568707043	93.922
Azerbaijan	1.38972636	7891.3131	2.797289382	95.090
Bosnia and Herzegovina	-0.89719405	5329.6350	1.148038011	72.483
Barbados	1.76950322	16179.5840	0.018086453	87.830
Bangladesh	6.99163889	1118.8537	6.061059359	95.589
Belgium	0.34000283	47355.3120	1.254676984	91.477
Burkina Faso	-0.25808952	703.8201	4.326837357	93.521
Bulgaria	-1.41818380	7864.7607	1.837521148	88.576
Bahrain	2.64629146	24989.4001	4.349842330	98.851
Burundi	4.40535234	274.8579	4.240651644	98.430
Benin	-1.00604387	943.6746	6.351831941	97.459
Brunei Darussalam	-0.20710873	41726.7840	-2.349746750	93.035
Bolivia	5.76660075	3081.8788	5.460567154	97.993
Brazil	6.32915223	12112.5903	0.503955740	93.330
Bahamas, The	1.50699859	29563.7463	-0.147988573	86.200
Bhutan	8.27106094	2749.3527	5.745455168	97.370

Análise de Agrupamento

Algoritmo K-means

Para usar o algoritmo **K-means** para agrupar os países, é necessário:

- Calcular a distância (dissimilaridade) entre os países;
- Escolher o número de grupos (ou clusters) a ser utilizado.

Para o cálculo da distância, temos um problema: as escalas das colunas são diferentes. Enquanto o PIB per Capita é dado em dólares por pessoa e os valores variam entre 255 e 116,613, os outros dados que selecionamos para a análise estão em porcentagem. Se não for feita uma transformação dos dados, o PIB per capita terá um peso muito maior no agrupamento do que os dados dos outros indicadores. Por isso, é necessário transformar todos os indicadores para uma escala única de média 0, que não afetará o resultado do agrupamento. Como a média de cada indicador será igual a zero (0), valores negativos indicarão que o indicador do país está abaixo da média, enquanto valores positivos indicarão que o indicador do país está acima da média.

```
dfi2014_escala <- scale(dfi2014)
# Conferindo o resultado para o Brasil
dfi2014_escala["Brazil", ]
##      Inflacao PIB_per_Capita Crescimento_PIB      Emprego
##  0.3204994   -0.1510604   -0.7782460    0.1843025
```

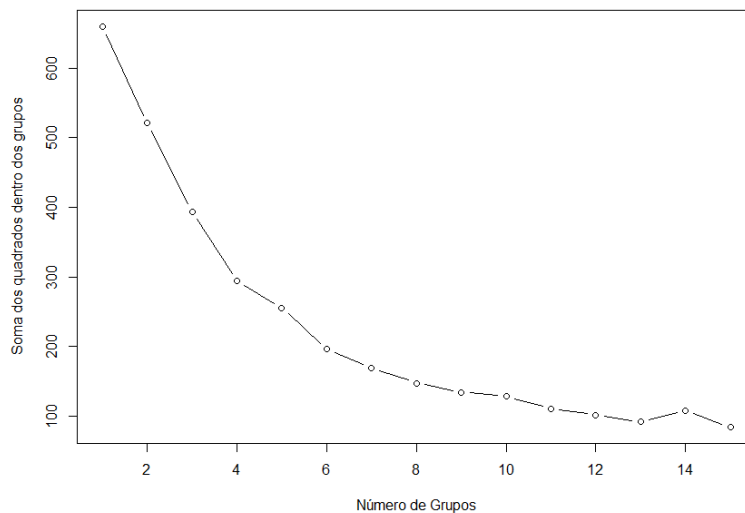
Na nova escala, temos que o Brasil apresenta inflação acima da média (maior do que zero), PIB per Capita abaixo da média (menor do que zero), Crescimento do PIB abaixo da média (menor do que zero) e Taxa de Emprego um pouco acima da média (maior do que zero).

Quantidade de Grupos

A determinação da quantidade de grupos não segue uma regra pré-definida e deve ser pesquisada pelo analista de dados. Cada projeto de agrupamento tem suas próprias particularidades. Contudo, alguns métodos analíticos podem ajudar nessa escolha, seja pela minimização da soma dos quadrados dos grupos ou pelo auxílio visual de um dendrograma.

Para determinar o número de grupos pelo primeiro método, observe o gráfico abaixo que é obtido com a execução dos comandos a seguir:

```
# referencia: http://www.statmethods.net/advstats/cluster.html
wss <- (nrow(df2014_escala)-1)*sum(apply(df2014_escala,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(df2014_escala,
  centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Número de Grupos",
  ylab="Soma dos quadrados dentro dos grupos")
```

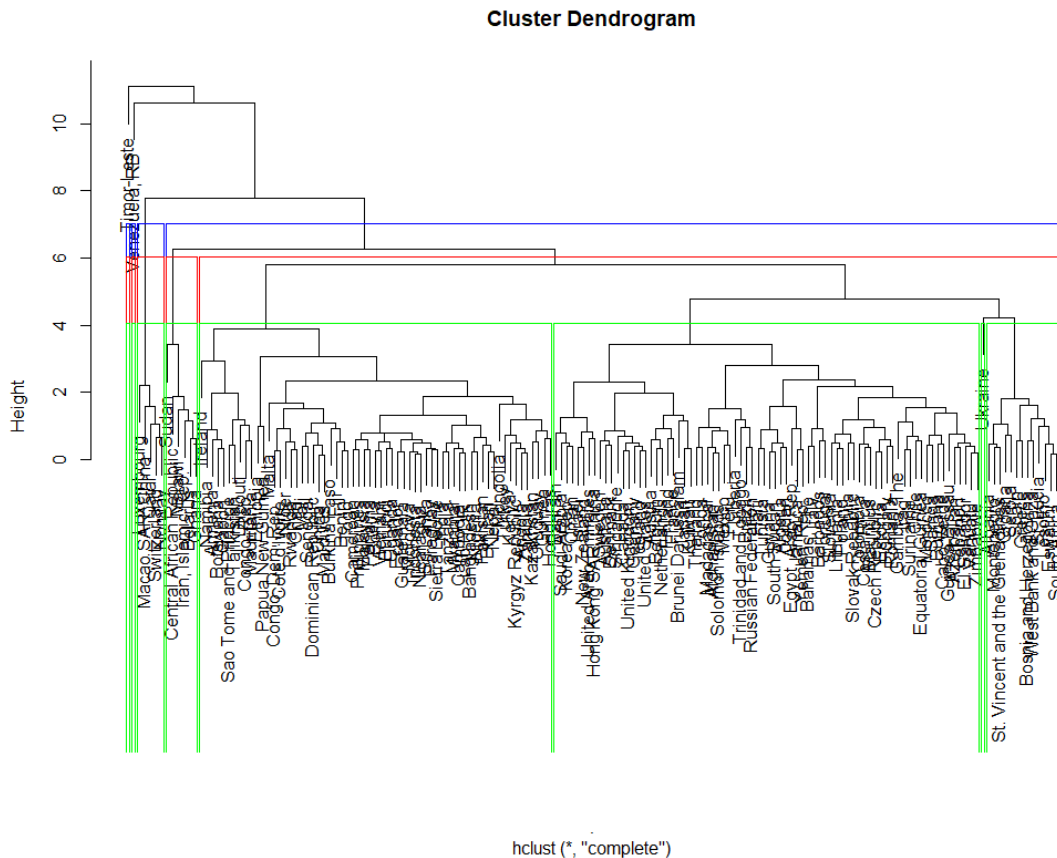


A soma dos quadrados dos grupos se mantém praticamente estável a partir de aproximadamente 8 segmentos ou grupos. Contudo, é preciso pensar qual a interpretação teríamos com oito grupos. Ou seja, podemos dividir os países em 8 grupos, mas parece um número alto para interpretarmos o que estes grupos significariam. Qual conhecimento seria obtido por meio desses 8 grupos?

Pelo segundo método, criaremos um dendrograma para analisar a distribuição hierárquica dos grupos considerando 4 grupos (azul), 5 grupos (vermelho) ou 8 grupos (verde):

```
dendo <- df2014_escala %>% dist %>% hclust
plot(dendo)
rect.hclust(dendo, k = 4, border = "blue")
rect.hclust(dendo, k = 5, border = "red")
rect.hclust(dendo, k = 8, border = "green")
```

Como você poderá observar, a visualização do dendrograma não é muito fácil, pois no último nível são apresentados todos os países. Por isso, pintamos de cores os diferentes níveis de grupos para facilitar a interpretação de qual seria um número adequado de grupos para análise.



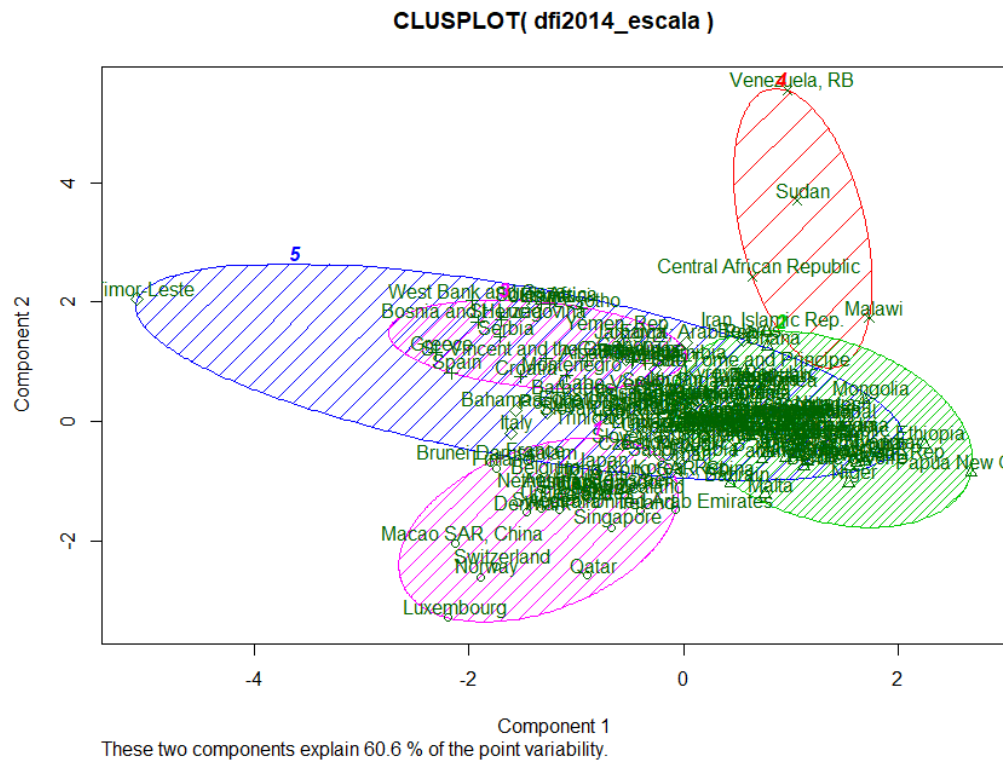
A posição de cada país no dendrograma é determinada pela dissimilaridade entre cada um dos outros países. Analisando o resultado do dendrograma, 5 parece ser uma boa escolha para a quantidade de grupos para o modelo desta análise.

Análise dos Resultados

Uma outra possibilidade de visualização gráfica dos grupos, que facilita a análise e interpretação dos resultados, pode ser obtida através das instruções a seguir. Neste caso os grupos são apresentados considerando as variáveis que mais os discriminam.

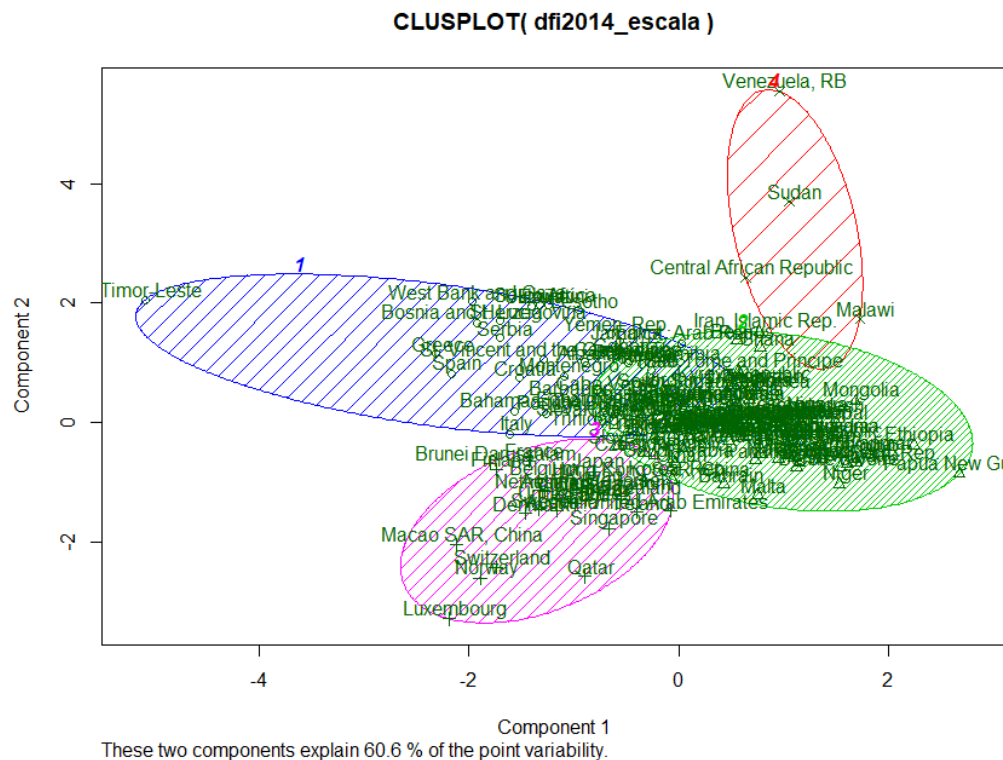
```
library(cluster)
library(fpc)
grupos <- kmeans(df2014_escala, centers=5)
clusplot(df2014_escala, grupos$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```

Note que como os gráficos são bidimensionais o algoritmo automaticamente escolhe os dados de componentes que explicam a variabilidade dos objetos para representá-los.

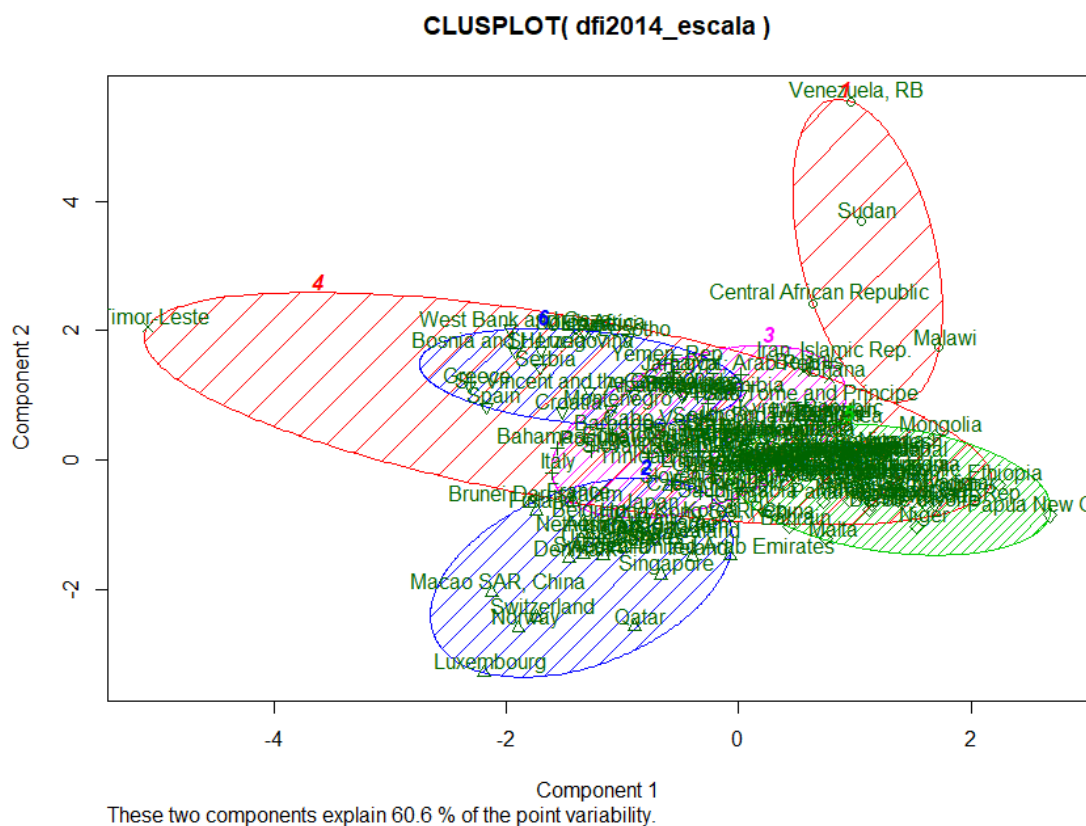


Você pode agora refazer a análise alterando o número de grupos ou clusters e visualizar os diferentes resultados possíveis para o agrupamento. Por exemplo, reduzindo o número de clusters para 4 teríamos o seguinte resultado:

```
grupos <- kmeans(dfi2014_escala, centers=4)
clusplot(dfi2014_escala, grupos$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```



Agora, repetindo as instruções, só que aumentando o número de clusters para 6, teríamos o seguinte resultado:



Para efeito de nossa análise vamos considerar 5 grupos ou clusters de países. Porém você poderia discordar e considerar um número diferente, desde que tenha uma argumentação analítica razoável para isso. Conforme comentamos a análise de agrupamento é subjetiva e depende de uma adequada interpretação e argumentação do analista.

Considerando, portanto, 5 grupos de países, as instruções a seguir mostram a distância entre o Brasil e alguns outros países (você pode escolher outros países, modificando as instruções, se desejar):

```
dfi2014_escala[c("Brazil", "Chile", "Colombia", "Norway", "United States"),] %>% dist
```

	Brazil	Chile	Colombia	Norway				
	Brazil	Chile	Colombia	Mexico	Norway	France	United States	
Chile	0.4429654							
Colombia	1.3364965	0.9742668						
Mexico	0.7945573	0.4703161	0.8542182					
Norway	4.0366891	3.8681485	4.2711430	4.0066078				
France	1.7839098	1.5909551	1.9776061	1.8921756	2.7733952			
United States	2.1741188	1.9315990	2.3004293	2.0825910	2.0031443	0.9934172		
Canada	2.0179168	1.7496472	2.0632350	1.9026028	2.2300525	0.8823924	0.2616168	

Podemos ver, por exemplo, que o Brasil tem uma distância euclidiana de 0,4430 em relação ao Chile, 2,1741 em relação aos Estados Unidos e 4,0367 em relação à Noruega. Ou seja, levando em conta os indicadores macroeconômicos considerados nesta análise, conforme esperado, é possível dizer que o Brasil é mais similar com países da América Latina do que com países ricos da América do Norte e Europa.

Podemos também ver qual a distribuição do grau de dissimilaridade do Brasil com o resto do mundo, solicitando a comparação com países com menor e maior dissimilaridade:

```
mat_brasil <- dfi2014_escala %>% dist(diag = TRUE, upper = TRUE) %>% as.matrix
# 5 países com MENOR dissimilaridade
mat_brasil[, "Brazil"] %>% sort() %>% head(6)
  Brazil Russian Federation      Chile      Suriname Equatorial Guinea      Iraq
0.0000000      0.3542751 0.4429654  0.4662700      0.5495528  0.6962494
# 5 países com MAIOR dissimilaridade
mat_brasil[, "Brazil"] %>% sort() %>% tail(5)
  Norway      Sudan Luxembourg Timor-Leste      Venezuela, RB
4.036689 4.781816  5.112100  7.438519      8.477183
```

O resultado dos 5 países mais distantes do Brasil é curioso: dentre eles, há 2 países ricos (Noruega e Luxemburgo) e 3 pobres (Sudão, Timor-Leste e Venezuela). Como você interpretaria estes resultados?

Vamos agora criar os segmentos, ou seja, os grupos e seus respectivos países:

```
# fixar uma seed (semente) para garantir a reprodutibilidade da análise:
set.seed(123)
# criar os clusters ou grupos
lista_clusteres <- kmeans(dfi2014_escala, centers = 5)$cluster
# função customizada para calcular a média dos indicadores para cada cluster
cluster.summary <- function(data, groups) {
  x <- round(aggregate(data, list(groups), mean), 2)
  x$qtd <- as.numeric(table(groups))
  # colocar coluna de quantidade na segunda posição
  x <- x[, c(1, 6, 2, 3, 4, 5)]
  return(x)
}
(tabela <- cluster.summary(dfi2014, lista_clusteres))
  Group.1 qtd Inflacao PIB_per_Capita Crescimento_PIB Emprego
1      1   4   37.04      4753.90      1.15   91.77
2      2  29    2.78      9986.94    0.33  82.25
3      3  27    1.57     58070.44    2.23  94.36
4      4  46    3.80      3761.52    6.53  96.50
5      5  60    4.17      8405.69    2.87  92.85
```

Para melhorar a apresentação visual do resultado acima, usaremos o pacote **formattable** com uma função para colorir de verde o valor caso seja superior ou igual à média do indicador e vermelho em caso contrário.

```
colorir.valor <- function(x) ifelse(x >= mean(x), style(color = "green"), style(color = "red"))

nome_colunas <- c("Cluster", "Quantidade de países do Grupo", "Taxa de Inflação (%)",
  "PIB Per Capita (US$)", "Crescimento anual do PIB (%)", "Taxa de Emprego (%)")
formattable(
  tabela,
  list(
    pib_per_capita = formatter("span", style = x ~ colorir.valor(x)),
    crescimento_pib = formatter("span", style = x ~ colorir.valor(x)),
    emprego = formatter("span", style = x ~ colorir.valor(x))
  ), col.names = nome_colunas, format = "markdown", pad = 0
)
```

Interpretação dos Grupos

Considerando, então, 5 grupos de países distintos, obtemos os seguintes resultados médios para cada um dos grupos:

Cluster	Quantidade de países do Grupo	Taxa de Inflação (%)	PIB Per Capita (US\$)	Crescimento anual do PIB (%)	Taxa de Emprego (%)
1	4	37.04	4753.90	1.15	91.77
2	29	2.78	9986.94	0.33	82.25
3	27	1.57	58070.44	2.23	94.36
4	46	3.80	3761.52	6.53	96.50
5	60	4.17	8405.69	2.87	92.85

Podemos finalmente analisar estes resultados e interpretar e rotular cada um dos 5 grupos ou clusters obtidos, de acordo com as características dos países de cada grupo, conforme o exemplificado a seguir:

- **Cluster 1 (países em crise):** inflação muito alta, PIB per capita baixo, crescimento baixo, emprego baixo;
- **Cluster 2 (países pobres):** inflação baixa, PIB per capita médio, crescimento baixo, emprego baixo;
- **Cluster 3 (países ricos):** inflação baixa, PIB per capita muito alto, crescimento médio, emprego alto;
- **Cluster 4 (países em desenvolvimento acelerado):** inflação alta, PIB per capita muito baixo, crescimento alto, emprego muito alto;
- **Cluster 5 (países em desenvolvimento lento):** inflação alta, PIB per capita médio, crescimento médio, emprego baixo.

Para finalizar, vamos determinar qual é o cluster do Brasil e quais os outros países estão no mesmo grupo?

```
dfi2014$cluster <- lista_clusteres
dfi2014["Brazil",]
      Inflacao PIB_per_Capita Crescimento_PIB Emprego cluster
Brazil 6.329152    12112.59      0.5039557   93.33      5
cl_brasil <- dfi2014["Brazil", ]$cluster
x <- dfi2014[dfi2014$cluster == cl_brasil, ]
x[order(-x$PIB_per_Capita),] %>% knitr::kable()
```

	Inflacao	PIB_per_Capita	Crescimento_PIB	Emprego	cluster
Afghanistan	4.67399604	613.8563	2.7245434	98.265	5
Angola	7.28038730	5408.4105	4.8226255	92.571	5
Azerbaijan	1.38972636	7891.3131	2.7972894	95.090	5
Burkina Faso	-0.25808952	703.8201	4.3268374	93.521	5
Bulgaria	-1.41818380	7864.7607	1.8375211	88.576	5
Brazil	6.32915223	12112.5903	0.5039557	93.330	5
Belarus	18.11955435	8318.5127	1.7263849	94.107	5
Chile	4.71867528	14670.9972	1.7667398	93.335	5

	Inflacao	PIB_per_Capita	Crescimento_PIB	Emprego	cluster
Colombia	2.89781874	8114.0843	4.7283122	91.428	5
Costa Rica	4.51934650	10547.1518	3.5153386	90.941	5
Czech Republic	0.34398859	19744.5586	2.7151161	93.892	5
Algeria	2.91692692	5493.0568	3.7891212	89.793	5
Ecuador	3.58922017	6377.0915	3.7888685	96.520	5
Estonia	-0.10617515	20247.1993	2.8885638	92.648	5
Egypt, Arab Rep.	10.14457120	3378.8314	2.9159119	86.895	5
Ghana	15.48961603	1968.8583	2.8974388	93.523	5
Gambia, The	5.94737492	622.0541	-0.9402355	90.448	5
Guinea	9.71397733	787.2381	3.7074516	95.767	5
Equatorial Guinea	4.30999866	19368.2309	0.4150663	91.468	5
Guinea-Bissau	-1.50924461	622.4817	0.9645608	95.725	5
Guyana	0.85263050	4031.6007	3.8469239	87.633	5
Honduras	6.12924930	2190.6507	3.0580806	94.512	5
Hungary	-0.22756627	14197.8370	4.2246538	92.275	5
Iraq	2.23597408	6818.8046	0.7000000	92.075	5
Iran, Islamic Rep.	17.22132871	5608.6025	4.6034189	89.430	5
Jordan	2.89893170	4047.5531	3.0963303	88.100	5
Kenya	6.87815499	1315.8045	5.3571256	90.412	5
Kyrgyz Republic	7.53424730	1279.7698	4.0000000	91.950	5
Korea, Rep.	1.27471470	27811.3664	3.3414478	96.500	5
Kazakhstan	6.84944977	12807.2607	4.2000000	94.940	5
Lebanon	1.85460421	7712.0628	1.8842857	93.720	5
Liberia	9.86111286	721.1828	0.7011439	97.793	5
Lithuania	0.10378991	16545.1227	3.5375858	89.302	5
Latvia	0.62049064	15716.3691	1.8582437	89.154	5

	Inflacao	PIB_per_Capita	Crescimento_PIB	Emprego	cluster
Morocco	0.44231005	3171.6992	2.6694939	90.300	5
Mauritania	3.53436856	1366.8090	5.5795439	89.789	5
Mauritius	3.21769192	10153.9382	3.7445758	92.533	5
Mexico	4.01861608	10922.3760	2.8043401	95.191	5
Oman	1.02234314	20131.9813	2.7510316	96.151	5
Peru	3.24496304	6679.3419	2.3819383	97.038	5
Poland	0.05382131	14347.9146	3.3184454	91.010	5
Romania	1.06830988	10026.9736	3.4108091	93.198	5
Russian Federation	7.82016970	14100.7291	0.6999994	94.840	5
Saudi Arabia	2.24185349	24463.9032	3.6524817	94.280	5
Solomon Islands	5.16590238	1996.7812	2.2500907	97.903	5
Slovenia	0.19934383	24194.9744	2.9505652	90.332	5
Suriname	3.38341273	9472.0076	0.2555031	93.060	5
South Sudan	1.65522360	1258.3781	3.3744808	87.558	5
São Tome and Principe	6.99849944	1782.7978	6.5499331	86.114	5
El Salvador	1.14134468	3589.0406	1.7112698	95.837	5
Thailand	1.89514182	5951.8837	0.9844141	99.424	5
Tajikistan	6.10442765	1104.1717	6.7059682	88.351	5
Tonga	2.51087633	4393.9408	2.0726073	98.890	5
Turkey	8.85457271	12095.8546	5.1666907	90.120	5
Trinidad and Tobago	5.68441815	20169.6530	-0.9696842	97.776	5
Uruguay	8.87735333	16831.9729	3.2387912	93.453	5
Vanuatu	0.79886384	3088.2583	2.3310062	94.130	5
Samoa	-0.40681609	4188.7337	1.3967580	91.280	5
Zambia	7.80687554	1763.0562	4.6958264	92.293	5
Zimbabwe	-0.21293990	1434.8993	2.3769293	94.482	5

Conclusão

Você facilmente pode perceber que existem problemas com este resultado: por exemplo, no mesmo grupo, estão presentes a Coreia do Sul e países como Haiti e Zimbábue. Isso pode ser explicado por uma série de razões, tais como:

- O número e perfil dos indicadores macroeconômicos escolhidos não é bom o suficiente para determinar uma segmentação eficiente dos países, ou seja, precisaríamos melhorar a nossa análise acrescentando outros dados ou variáveis que caracterizam melhor os países;
- O número de grupos talvez devesse ser maior, porém teríamos alguma dificuldade em rotulá-los;
- O problema também pode ser atribuído a um erro aleatório, também chamado de ruído, do algoritmo K-means, uma vez que nenhum modelo é perfeito.
- E assim por diante.

Entretanto, apesar destas e outras deficiências, a análise de agrupamento fornece um conhecimento muito interessante sobre os objetos em análise, uma vez que os agrupa de acordo com suas semelhanças.

Agora é a sua Vez

Repita o mesmo exercício para os indicadores de um ano mais recente, 2020 por exemplo, e compare o que aconteceu com a posição do Brasil e de outros países de sua escolha em relação ao agrupamento obtido em 2014. Como você explicaria esse resultado comparativo?

Evidências da Realização deste Laboratório

Como resultado deste laboratório você deve entregar todas as saídas (relações e gráficos) geradas pelo modelo de análise. Seguindo a mesma sequência do laboratório, sugerimos que você crie um documento no Word e cole nele cada resultado obtido. Após cada resultado faça alguns comentários e argumente o número de grupos que você considerou mais adequado para agrupar os países.

Referências

- <https://sillasgonzaga.github.io/2016-06-28-clusterizacaoPaíses/>
- <https://databank.worldbank.org/reports.aspx?source=world-development-indicators>
- <https://stats.stackexchange.com/questions/31083/how-to-produce-a-pretty-plot-of-the-results-of-k-means-cluster-analysis>

Laboratório Agrupamento Hierárquico

O agrupamento ou clustering é a forma mais comum de aprendizado não supervisionado, um tipo de algoritmo de aprendizado de máquina usado para desenhar inferências a partir de dados não rotulados ou categorizados.

Neste segundo laboratório, você aprenderá a executar clusters hierárquicos em um conjunto de dados utilizando a linguagem R. Mais especificamente, você aprenderá sobre:

- O que é clustering, quando ele é usado e seus tipos.
- Como pré-processar seus dados.
- O funcionamento do algoritmo de clustering hierárquico em detalhes.
- Como executar a análise de cluster.
- Comparação com K-means.

Como o próprio nome sugere, os algoritmos de clustering agrupam um conjunto de pontos de dados em subconjuntos ou clusters. O objetivo dos algoritmos é criar clusters que sejam coerentes internamente, mas claramente diferentes uns dos outros externamente. Em outras palavras, as entidades dentro de um cluster devem ser tão semelhantes quanto possível entre si, e tão diferentes quanto possível das entidades dentro de outro cluster.

De um modo geral, existem duas formas de clustering de pontos de dados com base na estrutura e operação algorítmicas, denominados aglomerativa e divisiva.

- **Aglomerativo:** um método aglomerativo começa com cada observação em um conjunto distinto (singleton), e funde sucessivamente aglomerados juntando objetos até que um critério de parada esteja satisfeito.
- **Divisivo:** um método divisivo começa com todos os objetos em um único cluster e executa a divisão até que um critério de parada seja atendido.

Neste laboratório, você vai se concentrar na abordagem aglomerativa ou *bottom-up*, onde você começa com cada ponto de dados como seu próprio cluster e, em seguida, combina cada cluster com base em alguma medida de similaridade. A mesma ideia, só que de maneira inversa, pode facilmente ser adaptada para métodos divisivos.

A semelhança entre os clusters é muitas vezes calculada a partir das medidas de dissimilaridade, como a distância euclidiana entre dois clusters. Assim, quanto maior a distância entre dois clusters, melhor é o resultado.

Há muitas métricas de distância que você pode considerar para calcular a medida de dissimilaridade, e a escolha depende do tipo de dados contido no conjunto de dados em análise. Por exemplo, se você tem valores numéricos contínuos em seu conjunto de dados, você pode usar a distância euclidiana, se o dado for binário você pode utilizar a distância Jaccard (útil quando você está lidando com dados categóricos para clustering depois de ter aplicado uma codificação). Outras medidas de distância incluem Manhattan, Minkowski, Canberra etc.

Operações de pré-processamento para clustering

Há um algumas coisas que você deve cuidar antes de começar.

Escala

É imperativo que você normalize sua escala de valores de atributos antes de iniciar o processo de clustering. Isso é importante porque os valores de atributos de cada observação são representados como coordenadas no espaço n-dimensional (n é o número de dimensões) e, em seguida, as distâncias entre essas coordenadas são calculadas. Se essas coordenadas não estiverem normalizadas, poderão levar a resultados falsos.

Por exemplo, suponha que você tenha dados sobre a altura e o peso de três pessoas: **A** (6ft, 75kg), **B** (6ft, 77kg), **C** (8ft, 75kg). Se você representar esses recursos em um sistema de coordenadas bidimensionais, altura e peso, e calcular a distância euclidiana entre eles, a distância entre os seguintes pares seria:

A-B: 2 unidades

A-C: 2 unidades

Bem, a métrica de distância diz que ambos os pares A-B e A-C são semelhantes, mas na realidade eles claramente não são! O par A-B é mais similar do que o par A-C. Portanto, é importante normalizar esses valores primeiro e, em seguida, calcular a distância.

Há várias maneiras de normalizar os valores de atributos, você pode considerar padronizar toda a escala de todos os valores de atributos ($x(i)$) entre $[0,1]$ (conhecido como normalização mín-máx) aplicando a seguinte transformação:

$$x(s) = x(i) - \min(x) / (\max(x) - \min(x))$$

Você pode usar a função **normalize()** do R para isso ou você pode escrever sua própria função como:

```
normalize <- function(x) {(x-min(x))/(Max(x)-min(x))}
```

Outro tipo de dimensionamento pode ser conseguido através da seguinte transformação:

$$x(s) = x(i) - \text{mean}(x) / \text{sd}(x)$$

Onde $\text{sd}(x)$ é o desvio padrão dos valores do objeto. Isso garantirá que sua distribuição de valores de objetos tenha a média 0 e um desvio padrão de 1. Você pode conseguir isso através da função `scale()` em R.

Imputação de valor ausente

Também é importante lidar com valores ausentes ou nulos em seu conjunto de dados de antemão. Há muitas maneiras de lidar com esses valores, um é removê-los ou substituí-los com média, mediana, modo ou usar algumas técnicas de regressão avançada. A linguagem R tem muitos pacotes e funções para lidar com tratamento de valores ausentes como `impute()`, `Amelia`, `Mice`, `Hmisc` etc. Você poderá ler sobre `Amelia` neste laboratório.

Algoritmo de clustering hierárquico

A operação de chave em clusters aglomerativos hierárquicos é combinar repetidamente os dois clusters mais próximos em um cluster maior. Há três questões-chave que precisam ser respondidas:

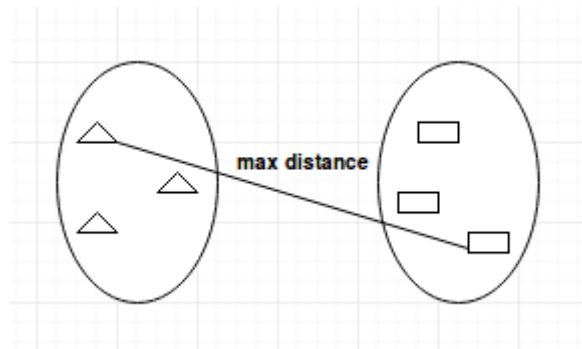
- Como você representa um cluster de mais de um ponto?
- Como você determina a "proximidade" de clusters?
- Quando você para de combinar clusters?

Esperamos que no final deste laboratório você seja capaz de responder a todas essas perguntas. Antes de aplicar clusters hierárquicos, vamos dar uma olhada no seu processo:

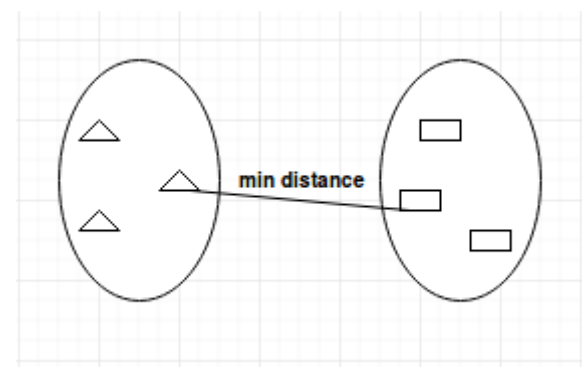
1. Ele começa calculando a distância entre cada par de pontos de observação e armazena em uma matriz de distância.
2. Em seguida, ele coloca todos os pontos em seu próprio cluster.
3. Depois, ele começa a mesclar os pares mais próximos de pontos com base nas distâncias da matriz de distância e, como resultado, a quantidade de clusters desce por 1.
4. Em seguida, ele recalcula a distância entre o novo cluster e os antigos e os armazena em uma nova matriz de distância.
5. Por fim, ele repete as etapas 2 e 3 até que todos os clusters sejam mesclados (aglomerados) em um único cluster.

Há várias maneiras de medir a distância entre clusters, a fim de decidir as regras para clustering, e eles são frequentemente chamados de métodos de ligação. Alguns dos métodos comuns de ligação são:

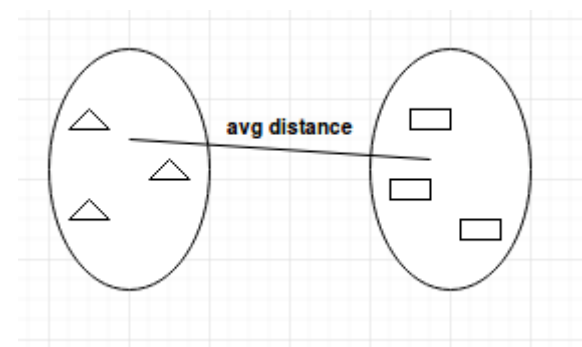
- **Vinculação completa:** calcula a distância máxima entre clusters antes da mesclagem.



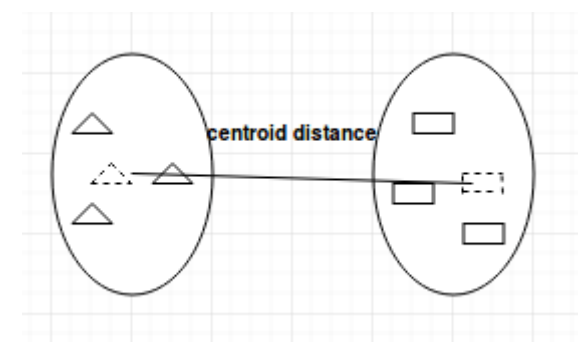
- **Ligação única:** calcula a distância mínima entre os clusters antes de mesclar. Essa ligação pode ser usada para detectar valores altos em seu conjunto de dados que podem ser outliers como eles serão mesclados no final.



- **Média-vinculação:** calcula a distância média entre clusters antes da mesclagem.



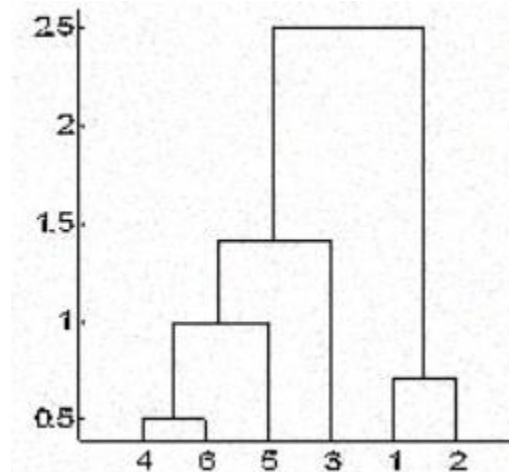
- **Ligação-Centroide:** localiza centroide do cluster 1 e centroide do cluster 2 e, em seguida, calcula a distância entre os dois antes de mesclar.



A escolha do método de ligação depende inteiramente de você e não há nenhum método ideal e rápido que lhe dará sempre bons resultados. Diferentes métodos de ligação levam a diferentes clusters.

Dendrograma

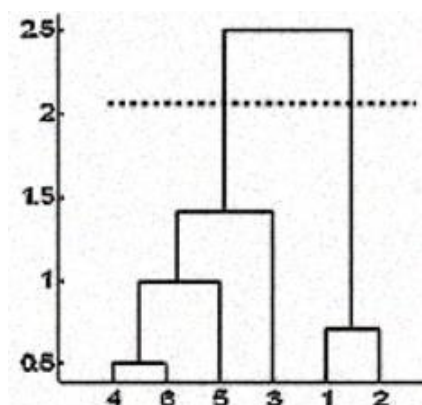
No clustering hierárquico, você categoriza os objetos em uma hierarquia semelhante a um diagrama de árvore, que é chamado de **dendrograma**. A distância de divisão ou mesclagem (chamada de altura) é mostrada no eixo y do dendrograma abaixo.



Na figura acima, nos primeiros 4 e 6 são combinados em um cluster, digamos que o cluster 1, uma vez que eles eram os mais próximos na distância seguido pelos pontos 1 e 2, digamos cluster 2. Depois disso, 5 foi mesclado no mesmo cluster 1 seguido por 3 resultando em dois clusters. Finalmente, os dois clusters são mesclados em um único cluster e é aqui que o processo de clustering é interrompido.

Uma pergunta que pode intrigá-lo é como você decide quando parar de mesclar os clusters? Bem, isso depende do conhecimento do domínio que você tem sobre os dados. Por exemplo, se você estiver aglomerando jogadores de futebol em um campo com base em suas posições no campo que representará suas coordenadas para o cálculo da distância, você já sabe que você deve terminar com apenas 2 clusters como só pode haver duas equipes jogando um jogo de futebol.

Mas às vezes você não tem uma informação como essa. Nesses casos, você pode aproveitar os resultados do dendrograma para aproximar o número de clusters. Você corta a árvore do dendrograma com uma linha horizontal em uma altura em que a linha pode atravessar a distância máxima para cima e para baixo sem cruzar o ponto de mesclagem. No caso acima seria entre alturas 1,5 e 2,5 como mostrado a seguir:



Se você fizer o corte como mostrado você vai acabar com apenas dois clusters.

Note que não é necessário fazer um corte apenas em tais lugares, você pode escolher qualquer ponto como o ponto de corte, dependendo de quantos clusters você deseja. Por exemplo, o corte abaixo de 1,5 e acima de 1 lhe dará 3 clusters.

Nota: esta não é uma regra rígida e rápida para decidir o número de clusters. Você também pode considerar parcelas como diagrama de silhueta, diagrama do cotovelo, ou algumas medidas numéricas, como o índice de Dunn, gama de Hubert, etc. que mostra a variação de erro com o número de clusters (k), e você escolhe, então, o valor de k onde o erro é menor.

Medindo a Qualidade dos Clusters

Talvez a parte mais importante em qualquer tarefa de aprendizagem não supervisionada seja a análise dos resultados. Depois de ter realizado o clustering usando um determinado algoritmo e quaisquer conjuntos de parâmetros, você precisa se certificar de que fez a análise corretamente. Mas como você determina isso?

Há muitas medidas para fazer isso, talvez o mais popular seja o índice de Dunn. O índice de Dunn é a razão entre as distâncias mínimas entre clusters e o diâmetro máximo do aglomerado intragrupo. O diâmetro de um aglomerado é a distância entre os dois pontos mais profundos. A fim de ter clusters bem separados e compactos, você deve procurar obter um índice de Dunn maior.

Agrupamento Hierárquico em Ação

Agora você vai aplicar o conhecimento que estivemos relatando até agora para resolver um problema do mundo real.

Você aplicará clustering hierárquico no conjunto de dados **seeds** (sementes). Este conjunto de dados consiste em medições de propriedades geométricas de grãos pertencentes a três variedades diferentes de trigo: Kama, Rosa e Canadense. Há variáveis que descrevem as propriedades das sementes como a área, o perímetro, o coeficiente da assimetria etc. Há 70 observações para cada variedade de trigo. Você pode encontrar os detalhes sobre o conjunto de dados para download no e-Disciplinas.

Comece importando o conjunto de dados em um dataframe com a função `read.csv()` do R.

Observe que o arquivo não tem cabeçalhos e é separado por tabulação. Para manter a reprodutibilidade dos resultados, você precisa usar a função `set.seed()`.

```
set.seed(786)
file_loc <- 'seeds.txt'
seeds_df <- read.csv(file_loc, sep = '\t', header = FALSE)
```

Como o conjunto de dados não tem nomes de coluna, você dará nome de colunas a partir da descrição dos dados.

```
feature_name <-
c('area', 'perimeter', 'compactness', 'length.of.kernel', 'width.of.kernal',
  'asymmetry.coefficient', 'length.of.kernel.groove', 'type.of.seed')
colnames(seeds_df) <- feature_name
```

É aconselhável reunir algumas informações básicas úteis sobre o conjunto de dados, como suas dimensões, tipos e distribuição, número de ausentes (NA), etc. Você fará isso usando as funções `str()`, `summary()` e `is.na()` do R.

```
str(seeds_df)
'data.frame':    221 obs. of  8 variables:
 $ area                : num  15.3 14.9 14.3 13.8 16.1 ...
 $ perimeter           : num  14.8 14.6 14.1 13.9 15 ...
 $ compactness         : num  0.871 0.881 0.905 0.895 0.903 ...
 $ length.of.kernel    : num  5.76 5.55 5.29 5.32 5.66 ...
 $ width.of.kernal     : num  3.31 3.33 3.34 3.38 3.56 ...
 $ asymmetry.coefficient : num  2.22 1.02 2.7 2.26 1.35 ...
 $ length.of.kernel.groove: num  5.22 4.96 4.83 4.8 5.17 ...
 $ type.of.seed        : num  1 1 1 1 1 1 1 5 NA 1 ...
```

```
summary(seeds_df)

      area      perimeter      compactness      length.of.kernel
width.of.kernal asymmetry.coefficient
Min.      : 1.00    Min.      : 1.00    Min.      :0.8081    Min.      :0.8189
Min.      :2.630    Min.      :0.7651
1st Qu.:12.11    1st Qu.:13.43    1st Qu.:0.8577    1st Qu.:5.2447    1st
Qu.:2.956    1st Qu.:2.6002
Median :14.13    Median :14.29    Median :0.8735    Median :5.5180
Median :3.245    Median :3.5990
Mean     :14.29    Mean     :14.43    Mean     :0.8713    Mean     :5.5639
Mean     :3.281    Mean     :3.6935
3rd Qu.:17.09    3rd Qu.:15.69    3rd Qu.:0.8877    3rd Qu.:5.9798    3rd
Qu.:3.566    3rd Qu.:4.7687
Max.     :21.18    Max.     :17.25    Max.     :0.9183    Max.     :6.6750
Max.     :5.325    Max.     :8.4560
NA's     :1        NA's     :9        NA's     :14        NA's     :11
NA's     :12        NA's     :11

length.of.kernel.groove  type.of.seed
Min.      :3.485          Min.      :1.000
1st Qu.:5.045          1st Qu.:1.000
Median :5.226          Median :2.000
Mean     :5.408          Mean     :2.084
3rd Qu.:5.879          3rd Qu.:3.000
Max.     :6.735          Max.     :5.439
NA's     :15            NA's     :15
```

```
any(is.na(seeds_df))

[1] TRUE
```

Observe que esse conjunto de dados tem todas as colunas como valores numéricos. Porém, como há valores ausentes neste conjunto de dados você precisa limpar antes de executar o algoritmo de clustering. Para isso vamos executar o comando abaixo:

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

Note que agora temos apenas 203 observações no conjunto de dados, pois foram eliminadas aquela que apresentavam valores ausentes (NA).

Mas as escalas das características são diferentes e você precisa normalizá-lo. Além disso, os dados são rotulados e você já tem as informações sobre qual a observação pertence a qual variedade de trigo.

Agora você armazenará os rótulos em uma variável separada e excluirá a coluna **Type.of.Seed** do seu conjunto de dados para fazer o clustering. Mais tarde você usará os rótulos verdadeiros para verificar a qualidade ou precisão do seu agrupamento.

```
seeds_label <- seeds_df$type.of.seed
seeds_df$type.of.seed <- NULL
str(seeds_df)
```

```
'data.frame':    203 obs. of  7 variables:
 $ area                : num  15.3 14.9 14.3 13.8 16.1 ...
 $ perimeter           : num  14.8 14.6 14.1 13.9 15 ...
 $ compactness         : num  0.871 0.881 0.905 0.895 0.903 ...
 $ length.of.kernel    : num  5.76 5.55 5.29 5.32 5.66 ...
 $ width.of.kernal     : num  3.31 3.33 3.34 3.38 3.56 ...
 $ asymmetry.coefficient : num  2.22 1.02 2.7 2.26 1.35 ...
 $ length.of.kernel.groove: num  5.22 4.96 4.83 4.8 5.17 ...
- attr(*, "na.action")= 'omit' Named int  8 9 37 38 63 64 73 112 141
142 ...
..- attr(*, "names")= chr  "8" "9" "37" "38" ...
```

Como você observará que você retirou a coluna de rótulo verdadeiro do seu conjunto de dados.

Agora você usará a função **scale()** do R para dimensionar todos os seus valores de coluna.

```
seeds_df_sc <- as.data.frame(scale(seeds_df))
summary(seeds_df_sc)
```

area	perimeter	compactness
length.of.kernel	width.of.kernal	
Min. : -1.4783	Min. : -1.6633	Min. : -2.6925
Min. : -1.67620		Min. : -1.6712
1st Qu.: -0.8824	1st Qu.: -0.8579	1st Qu.: -0.5927
1st Qu.: -0.81868		1st Qu.: -0.8461
Median : -0.1803	Median : -0.1670	Median : 0.1058
Median : -0.05894		Median : -0.2238
Mean : 0.0000	Mean : 0.0000	Mean : 0.0000
Mean : 0.00000		Mean : 0.0000
3rd Qu.: 0.8728	3rd Qu.: 0.9286	3rd Qu.: 0.6908
3rd Qu.: 0.79329		3rd Qu.: 0.8155
Max. : 2.1486	Max. : 2.0317	Max. : 2.0300
Max. : 2.03133		Max. : 2.3327
asymmetry.coefficient	length.of.kernel.groove	
Min. : -1.95774	Min. : -1.8280	
1st Qu.: -0.75803	1st Qu.: -0.7595	
Median : -0.05279	Median : -0.3892	
Mean : 0.00000	Mean : 0.0000	
3rd Qu.: 0.72357	3rd Qu.: 0.9320	
Max. : 3.15435	Max. : 2.2938	

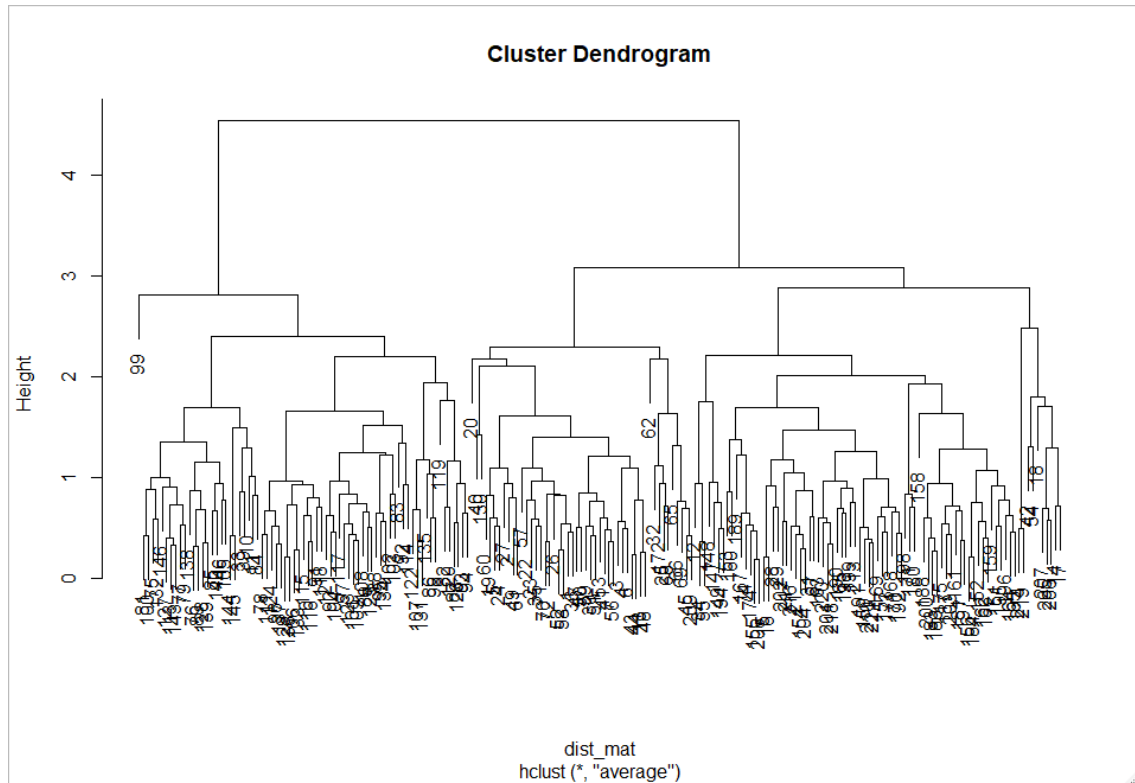
Observe que a média de todas as colunas é 0 e o desvio padrão é 1. Agora que você pré-processou seus dados, é hora de construir a matriz de distância. Como todos os valores aqui são valores numéricos contínuos, você usará o método de distância euclidiana.


```
dist_mat <- dist(seeds_df_sc, method = 'euclidean')
```

Neste ponto, você deve decidir qual método de ligação você deseja usar para fazer clustering hierárquico. Você pode tentar todos os tipos de métodos de ligação e, posteriormente, decidir sobre qual resultou num desempenho melhor. Aqui prosseguiremos com o método de ligação **average** (média).

Você construirá seu dendrograma plotando o objeto de cluster hierárquico que você construirá com a função **hclust()**. Você pode especificar o método de ligação por meio do argumento **Method**.

```
hclust_avg <- hclust(dist_mat, method = 'average')
plot(hclust_avg)
```



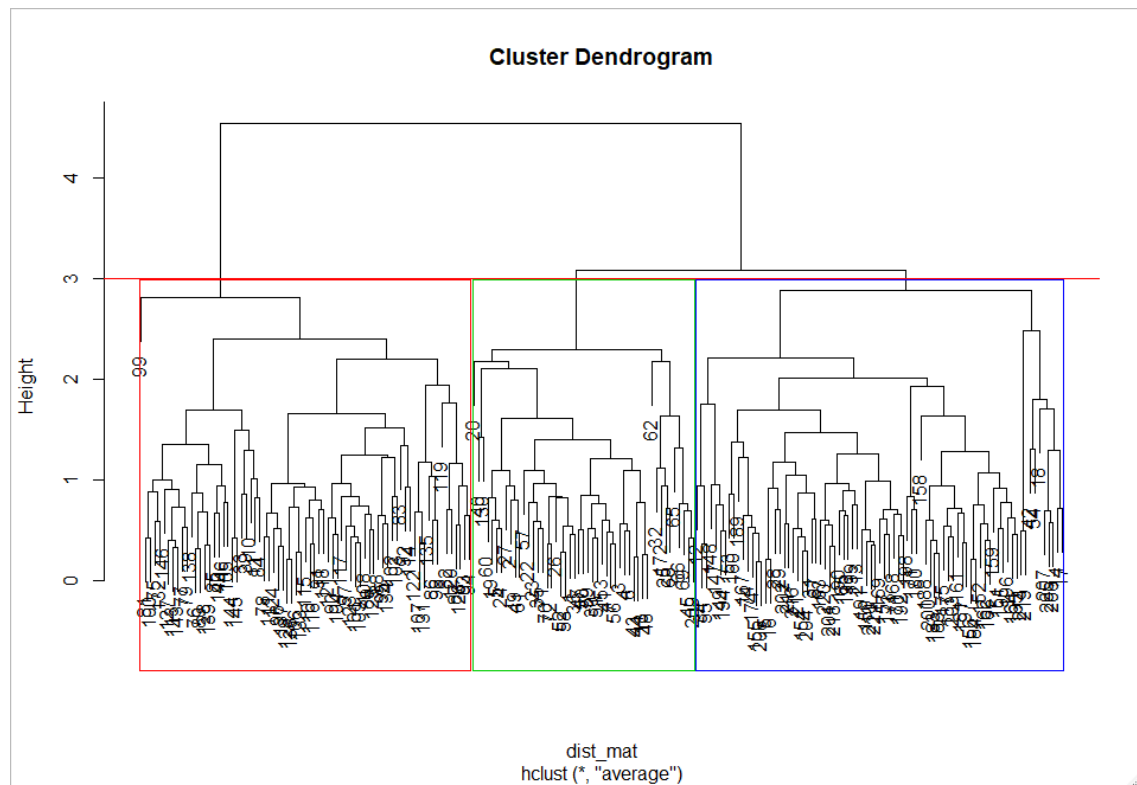
Observe como o dendrograma é construído e cada ponto de dados finalmente se funde em um único cluster com a altura (distância) mostrada no eixo y.

Em seguida, você pode cortar o dendrograma a fim criar o número desejado de grupos. Uma vez que neste caso você já sabe que pode haver apenas três tipos de trigo você vai escolher o número de clusters para ser $k = 3$, ou como você pode ver no dendrograma $h = 3$ você tem três clusters. Você usará a função **cutree()** do R para cortar a árvore com **hclust_avg** como um parâmetro e o outro parâmetro como $h = 3$ ou $k = 3$.

```
cut_avg <- cutree(hclust_avg, k = 3)
```

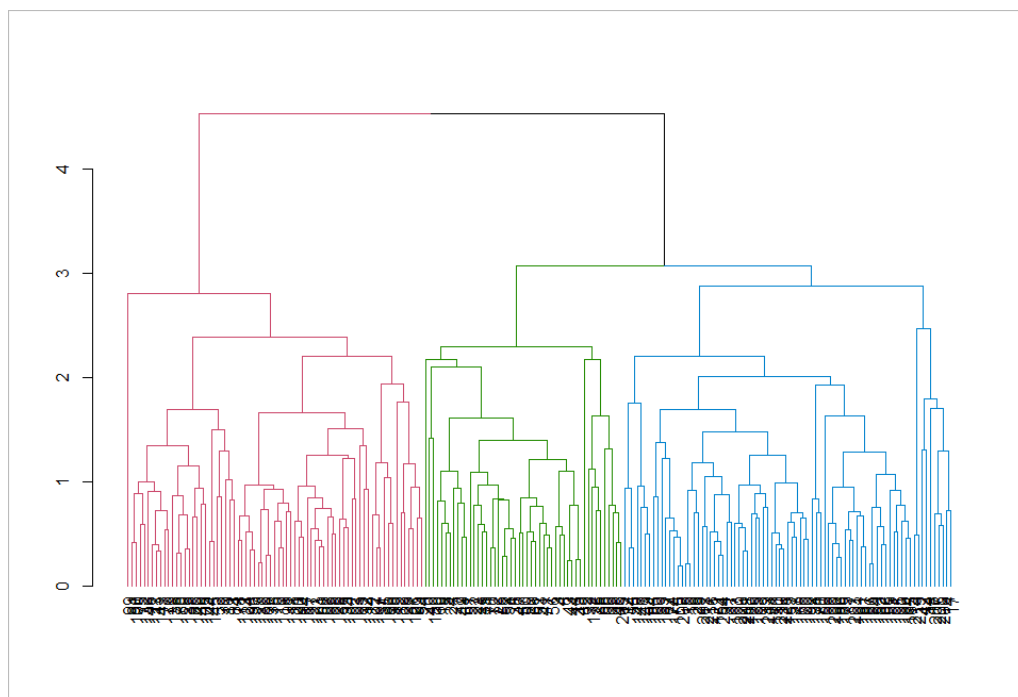
Se você deseja visualizar os clusters no dendrograma, pode usar a função **abline()** do R para desenhar a linha de corte e sobrepor compartimentos retangulares para cada cluster na árvore com a função **Rect.hclust()** conforme mostrado no código a seguir:

```
plot(hclust_avg)
rect.hclust(hclust_avg, k = 3, border = 2:6)
abline(h = 3, col = 'red')
```



Agora você pode ver os três clusters fechados em três caixas coloridas diferentes. Você também pode usar a função `color_branches()` da biblioteca `dendextend` para visualizar sua árvore com diferentes ramos coloridos.

```
install.packages("dendextend")
suppressPackageStartupMessages(library(dendextend))
avg_dend_obj <- as.dendrogram(hclust_avg)
avg_col_dend <- color_branches(avg_dend_obj, h = 3)
```



Agora você anexará os resultados do cluster obtidos de volta no dataframe original o nome da coluna o cluster com **mutate()**, do pacote **dplyr** e contará quantas observações foram atribuídas a cada cluster com a função **count()**.

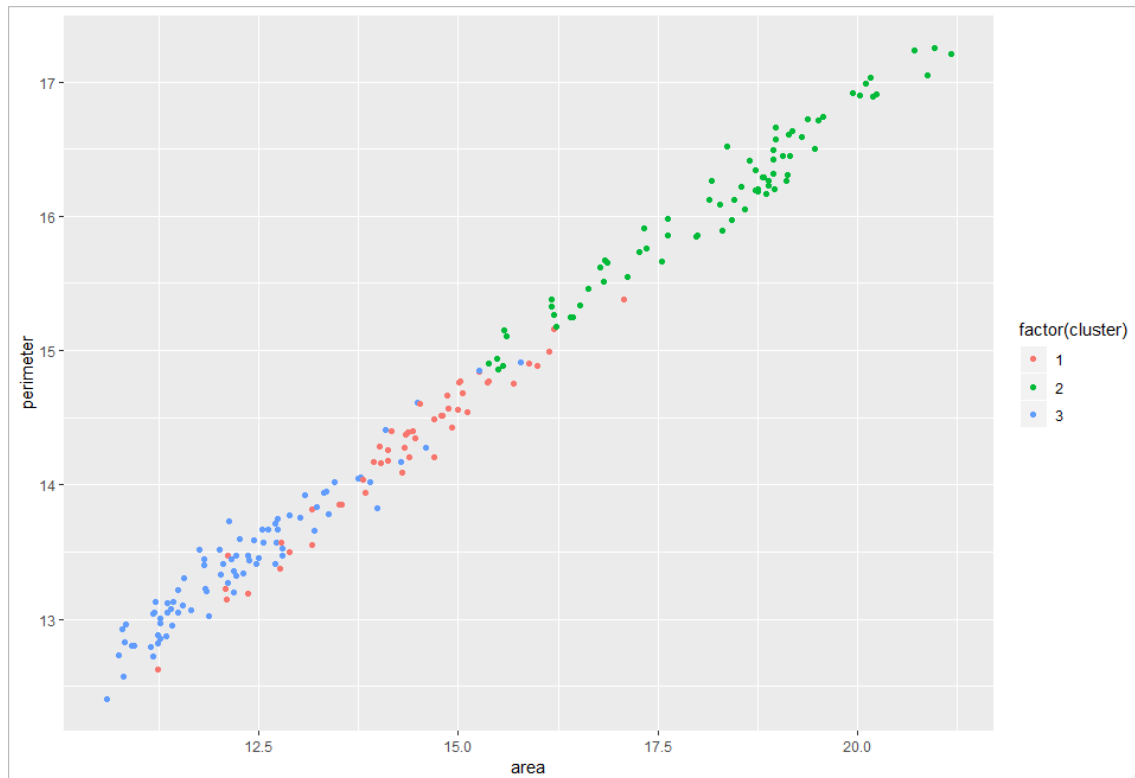
```
suppressPackageStartupMessages(library(dplyr))
seeds_df_cl <- mutate(seeds_df, cluster = cut_avg)
count(seeds_df_cl, cluster)
```

Você será capaz de ver quantas observações foram atribuídas em cada cluster. Note que na realidade a partir dos dados rotulados que você tinha 70 observações para cada variedade de trigo.

cluster	n
<int>	<int>
1	49
2	73
3	81

É comum avaliar a tendência entre dois recursos com base no clustering que você fez para extrair insights mais úteis do cluster de dados. Como um exercício, você pode analisar a tendência entre o perímetro do trigo e a área cluster-Wise com a ajuda do pacote ggplot2.

```
install.packages("ggplot2")
suppressPackageStartupMessages(library(ggplot2))
ggplot(seeds_df_cl, aes(x=area, y = perimeter, color =
factor(cluster))) + geom_point()
```



Observe que para todas as variedades de trigo parece haver uma relação linear entre o seu perímetro e área.

Como você já tem os rótulos verdadeiros para esse conjunto de dados, você também pode considerar a verificação cruzada dos resultados de clustering usando a função `table()`.

```
tabela (seeds_df_cl $cluster, seeds_label)
```

Se você tiver uma olhada na tabela que foi gerada, você verá claramente três grupos com 55 elementos ou mais. No geral, você pode dizer que seus clusters representam adequadamente os diferentes tipos de sementes, porque originalmente você tinha 70 observações para cada variedade de trigo. Os grupos maiores representam a correspondência entre os clusters e os tipos reais.

Observe que, em muitos casos, você realmente não tem os rótulos verdadeiros. Nesses casos, como já discutido, você pode ir para outras medidas como maximizar o índice de Dunn. Você pode calcular o índice de Dunn usando a função `dunn()` da biblioteca `clvalid`. Além disso, você pode considerar fazer a validação cruzada dos resultados, fazendo conjuntos de treinamento e teste, assim como você faz em qualquer outro algoritmo de aprendizado de máquina e, em seguida, fazendo o clustering quando você tem os rótulos verdadeiros.

Comparando com o algoritmo de clustering K-means

Há muitas diferenças fundamentais entre os dois algoritmos, embora qualquer um pode executar o agrupamento melhor do que o outro em casos diferentes. Algumas das diferenças são:

- **Distância usada:** o clustering hierárquico pode lidar virtualmente com qualquer métrica de distância, enquanto k-significa dependem de distâncias euclidianas.
- **Estabilidade de resultados:** k-means requer um passo aleatório na sua inicialização que pode produzir resultados diferentes se o processo é re-Run. Isso não seria o caso em clustering hierárquico.
- **Número de clusters:** enquanto você pode usar plotagens de cotovelo, enredo de silhueta etc. para descobrir o número certo de clusters em k-Means, hierárquico também pode usar todos aqueles, mas com o benefício adicional de alavancar o dendrograma para o mesmo.
- **Complexidade computacional:** K-means é menos dispendioso do que o clustering hierárquico e pode ser executado em grandes conjuntos de dados dentro de um período de tempo razoável, que é o principal motivo pelo qual k-means é mais popular.

Etapa Final: Relatório de Elaboração do Laboratório

Você deve entregar um relatório com os resultados das etapas elaboradas neste laboratório no e-Disciplinas, para formatá-lo siga estas orientações:

1. Crie um documento Word e identifique-o com o nome do laboratório, data de elaboração e o seu nome ou da dupla que o elaborou;
2. Crie um tópico para cada resultado que você considerar relevante (manipulação de dados ou resultado de algum processamento) identificando-o com um título e uma breve explicação. Os resultados podem ser imagens de gráficos gerados ou de listas de valores ou dados de resultados obtidos. Não devem ser incluídos os scripts ou instruções de processamento utilizados, inclua apenas os resultados que você considerar relevantes.
3. No final do relatório crie um último tópico denominado “Conclusões” e elabore comentários, sugestões e conclusões sobre o que você pode aprender com a elaboração deste laboratório.

Conclusão

Parabéns! Você concluiu este laboratório. Aprendeu a pré-processar seus dados, os fundamentos do clustering hierárquico e as métricas de distância e os métodos de ligação com que ele funciona, juntamente com seu uso em R. Você também sabe como o clustering hierárquico difere do algoritmo K-Means.

Referências

<https://www.datacamp.com/community/tutorials/hierarchical-clustering-R>