Algoritmos de Agrupamento - Aprendizado Não Supervisionado

Fabrício Jailson Barth

Pós Graduação - BandTec

Maio de 2015

Sumário

- Introdução e Definições
- Aplicações
- Algoritmos de Agrupamento
  - ⋆ Agrupamento Plano
  - ⋆ Agrupamento Hierárquico
- Considerações Finais

Introdução

Introdução e Definições

- Os algoritmos de agrupamento particionam um conjunto de objetos em agrupamentos.
- Normalmente, objetos são descritos e agrupados usando um conjunto de atributos e valores.
- **Não existe nenhuma informação sobre a classe ou categoria dos objetos**.

- Os algoritmos de agrupamento manipulam um conjunto de objetos. Este conjunto de objetos é chamado de **bags**.
- As **bags** permitem o aparecimento de múltiplos objetos com a mesma representação.
- **O objetivo dos algoritmos de agrupamento é colocar os objetos similares em um mesmo grupo e objetos não similares em grupos diferentes**.

Exemplo de dataset

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.10	3.50	1.40	0.20
2	4.90	3.00	1.40	0.20
3	4.70	3.20	1.30	0.20
4	4.60	3.10	1.50	0.20
5	5.00	3.60	1.40	0.20
6	5.40	3.90	1.70	0.40

Aplicações

- Agrupamento de objetos similares, onde **"objetos"** podem ser:
  - ⋆ agrupamento de documentos (textos) similares
  - ⋆ identificação de grupos em redes sociais
  - ⋆ segmentação de clientes
  - ⋆ pessoas - sistemas de recomendação
  - ⋆ palavras - processamento de linguagem natural
  - ⋆ identificação de plantas com características comuns
  - ⋆ entre outras coisas $\cdots$

Algoritmos

Algoritmos de Agrupamento

Existem dois tipos de estruturas produzidas por algoritmos de agrupamento:

- não hierárquicos ou **planos**
- agrupamentos **hierárquicos**

Agrupamento Plano

- Agrupamentos planos simplesmente contêm um certo número de agrupamentos e a **relação** entre os agrupamentos e geralmente **não-determinada**.
- A maioria dos algoritmos que produzem agrupamentos planos são **iterativos**.
- Eles iniciam com um conjunto inicial de agrupamentos e realocam os objetos em cada agrupamento de maneira iterativa.
- Até uma determinada **condição de parada**.

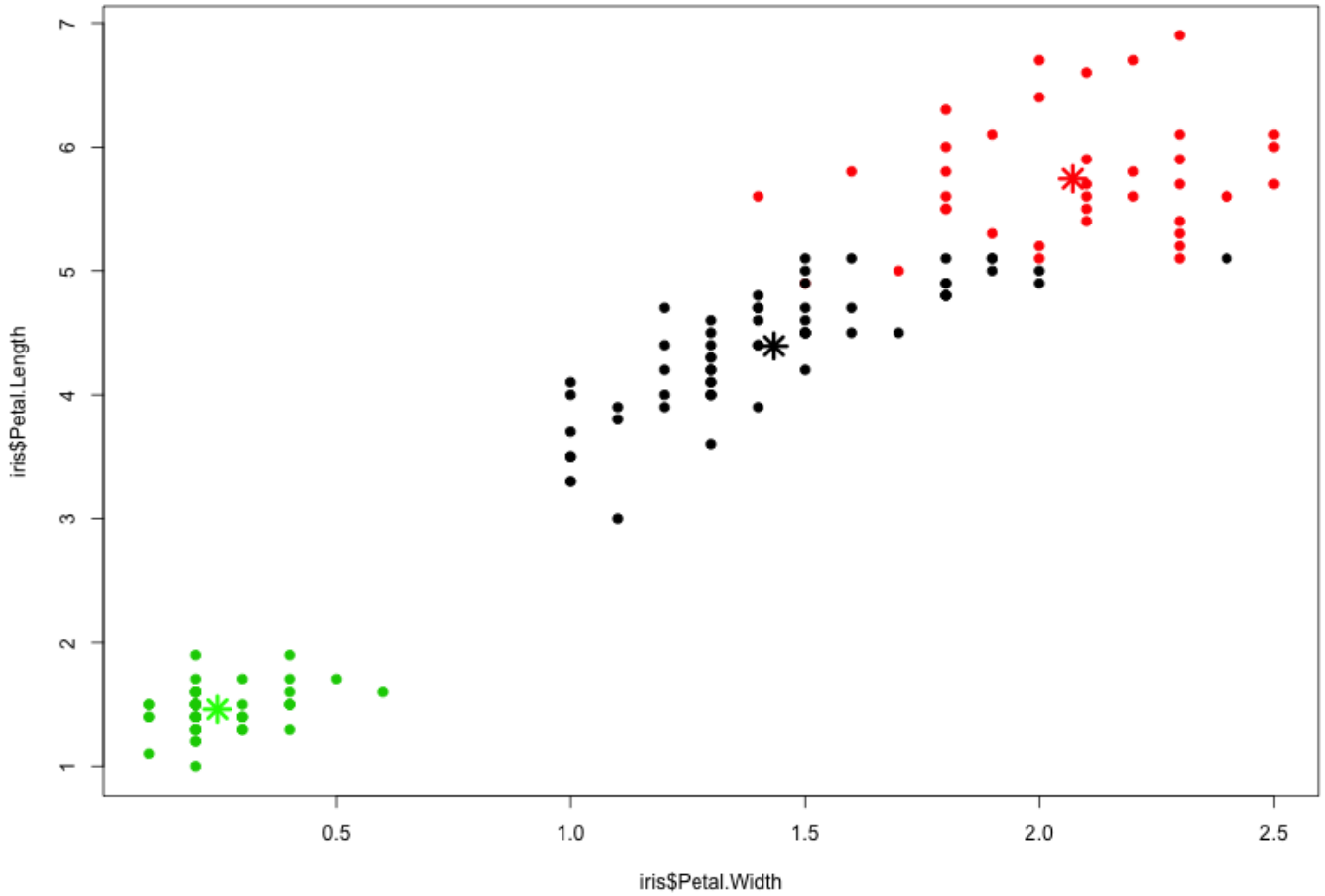
Agrupamentos **soft** e **hard**

Além da divisão entre os algoritmos hierárquicos e planos, tem-se a divisão entre os algoritmos **soft** e **hard**.

- Na abordagem **hard** cada objeto é inserido em um e somente um agrupamento.
- Na abordagem **soft** um objeto pode ser inserido em vários agrupamentos com diferentes níveis de pertinência.

Em agrupamentos hierárquicos, geralmente a abordagem é **hard**. Em agrupamentos planos, ambos os tipos de abordagens são comuns.

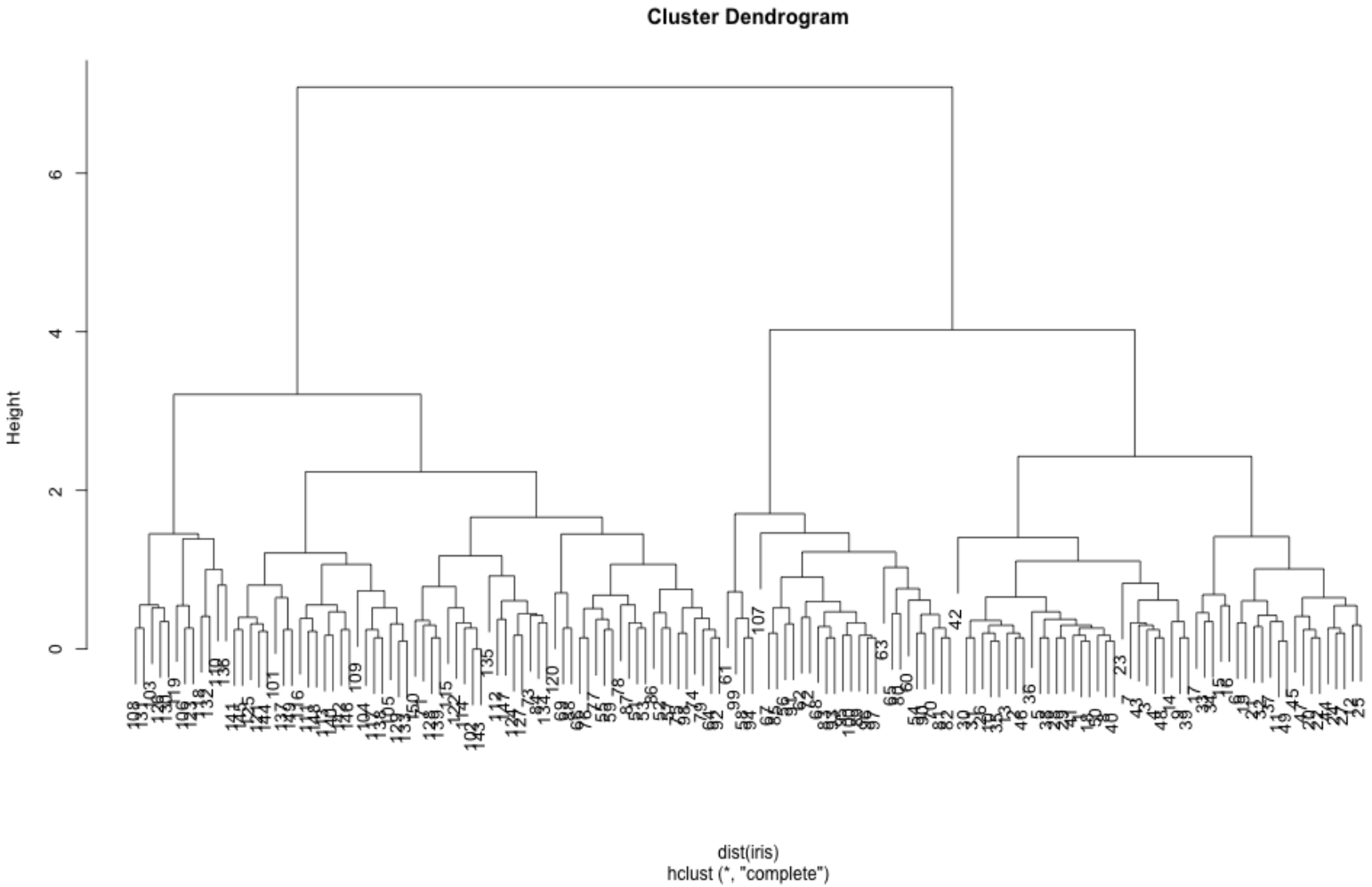
Agrupamento Plano *hard* (Exemplo)



Agrupamento Hierárquico

- Um agrupamento hierárquico é representado por uma árvore.
- Os nós folhas são os objetos.
- Cada nó intermediário representa o agrupamento que contêm todos os objetos de seus descendentes.

Agrupamento Hierárquico (Exemplo)



Agrupamento Hierárquico (Exemplo)

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
108	7.30	2.90	6.30	1.80
131	7.40	2.80	6.10	1.90
42	**4.50**	**2.30**	**1.30**	**0.30**

Algoritmos para Agrupamento Plano

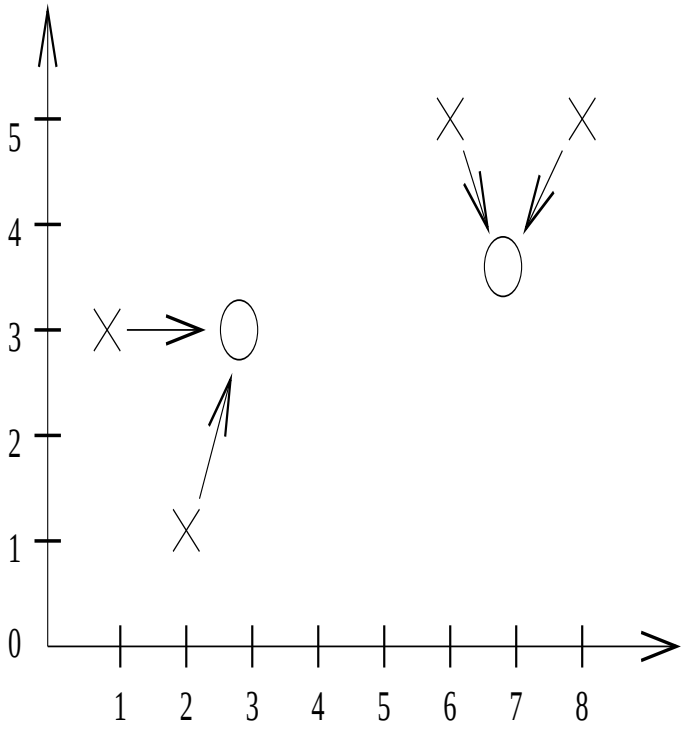
Algoritmos para agrupamento plano

- Utiliza diversas **iterações** para realocar os objetos nos melhores agrupamentos.
- **Critério de parada** é baseado na qualidade dos agrupamentos (similaridade média e cálculo para informação comum entre agrupamentos).
- É necessário determinar o **número de agrupamentos**:
  - ⋆ usando conhecimento à priori
  - ⋆ k, $k-1$ aumento significativo da qualidade, $k+1$ aumento reduzido da qualidade. Procurar por um k com este comportamento.

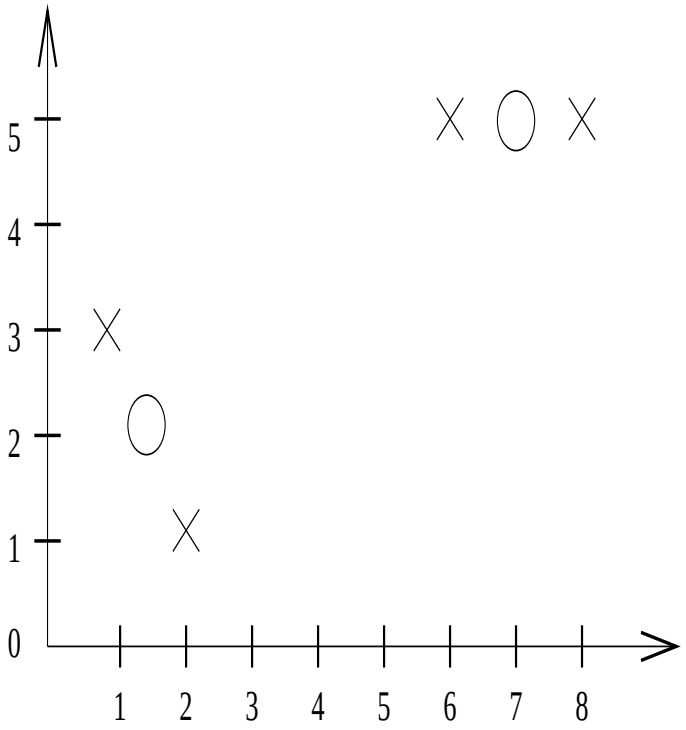
K-means

- Algoritmo de agrupamento **hard**
- Define o agrupamento pelo centro de massa dos seus membros.
- É necessário um conjunto inicial de agrupamentos.
- Seqüência de ações iterativas.
- Usualmente, diversas iterações são necessárias para o algoritmo convergir.

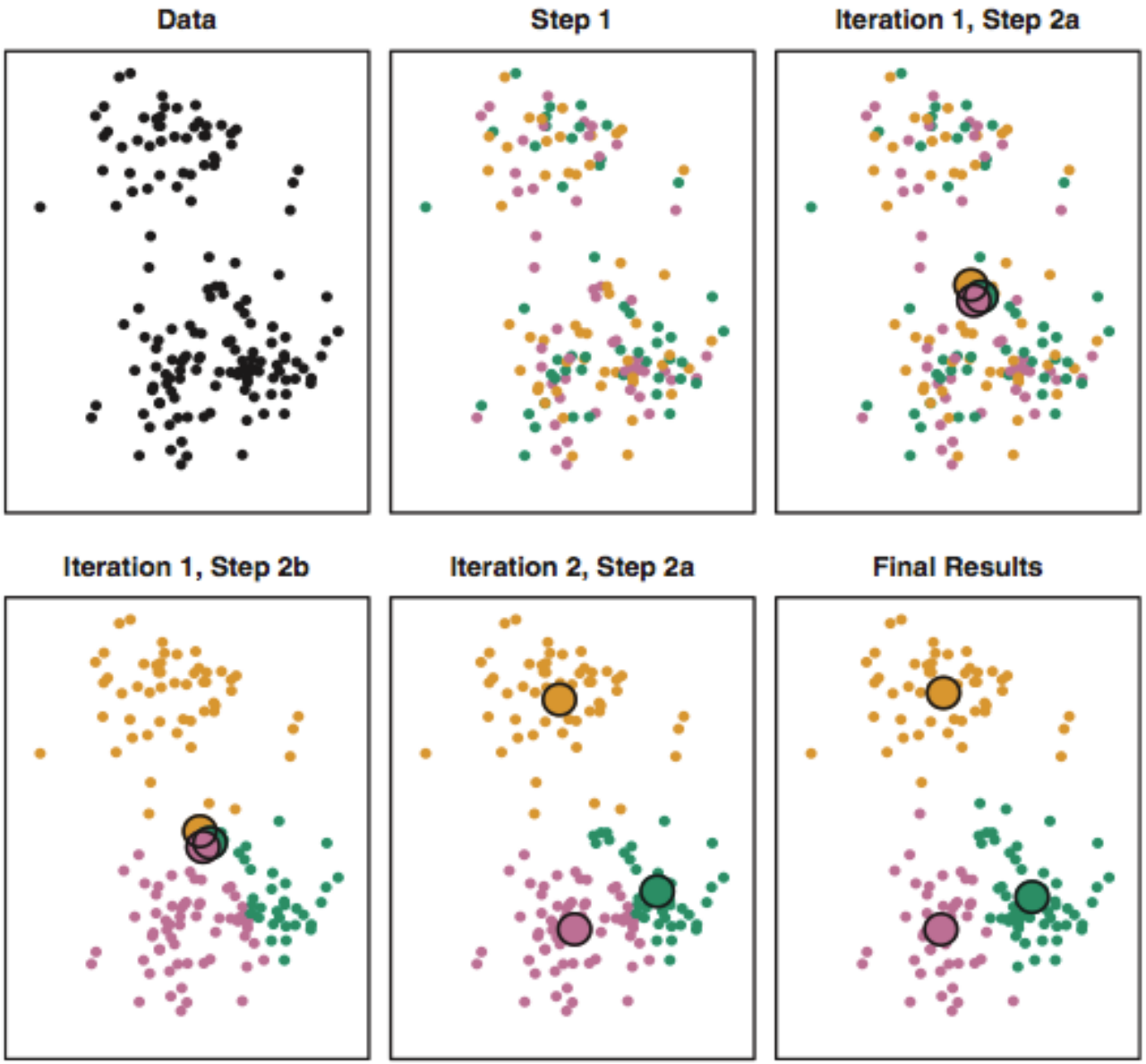
Iteração do algoritmo **K-means**



(1) Atribuição dos objetos aos agrupamentos



(2) Definição do centro do agrupamento


Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results

Algoritmo **K-means**

entrada: um conjunto $X = \{\overrightarrow{x_1}, \cdots, \overrightarrow{x_n}\} \subset \Re^m$

{conjunto inicial de agrupamentos}

uma medida de distância: d: $\Re^m \times \Re^m \rightarrow \Re$

uma função para computar o ponto central:

μ: $P(\Re) \rightarrow \Re^m$

selecionar k centros iniciais $\overrightarrow{f_1}, \cdots, \overrightarrow{f_k}$

while o critério de parada não for verdadeiro **do**

for todos os agrupamentos c_j **do**

$c_j = \{\overrightarrow{x_i} \mid \forall \overrightarrow{f_l}\, d(\overrightarrow{x_i}, \overrightarrow{f_j}) \leq d(\overrightarrow{x_i}, \overrightarrow{f_l})\}$ {os agrupamentos c_j recebem objetos levando-se em consideração o seu centro f_j}

end for

for todos os pontos centrais $\overrightarrow{f_j}$ **do**

$\overrightarrow{f_j} = \mu(c_j)$

end for

end while

Algoritmo **K-means**

- A medida de distância pode ser a distância Euclidiana:

$$| \overrightarrow{x} - \overrightarrow{y} | = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{1}$$

- a função para computar o ponto central pode ser:

$$\overrightarrow{\mu} = \frac{1}{M} \sum_{\overrightarrow{x} \in C} \overrightarrow{x} \tag{2}$$

onde M é igual ao número de pontos no agrupamento C.

Problema...

Iris Problem

- Considere uma base de dados sobre flores do gênero **Iris**.
- Esta base de dados possui informações sobre o **comprimento** e **largura** das **sépalas** e das **pétalas** das flores.

Blue Flag Iris - Dados

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.10	3.50	1.40	0.20
2	4.90	3.00	1.40	0.20
3	4.70	3.20	1.30	0.20
4	4.60	3.10	1.50	0.20
5	5.00	3.60	1.40	0.20
6	5.40	3.90	1.70	0.40

Todas as medidas são em cm.

Pergunta

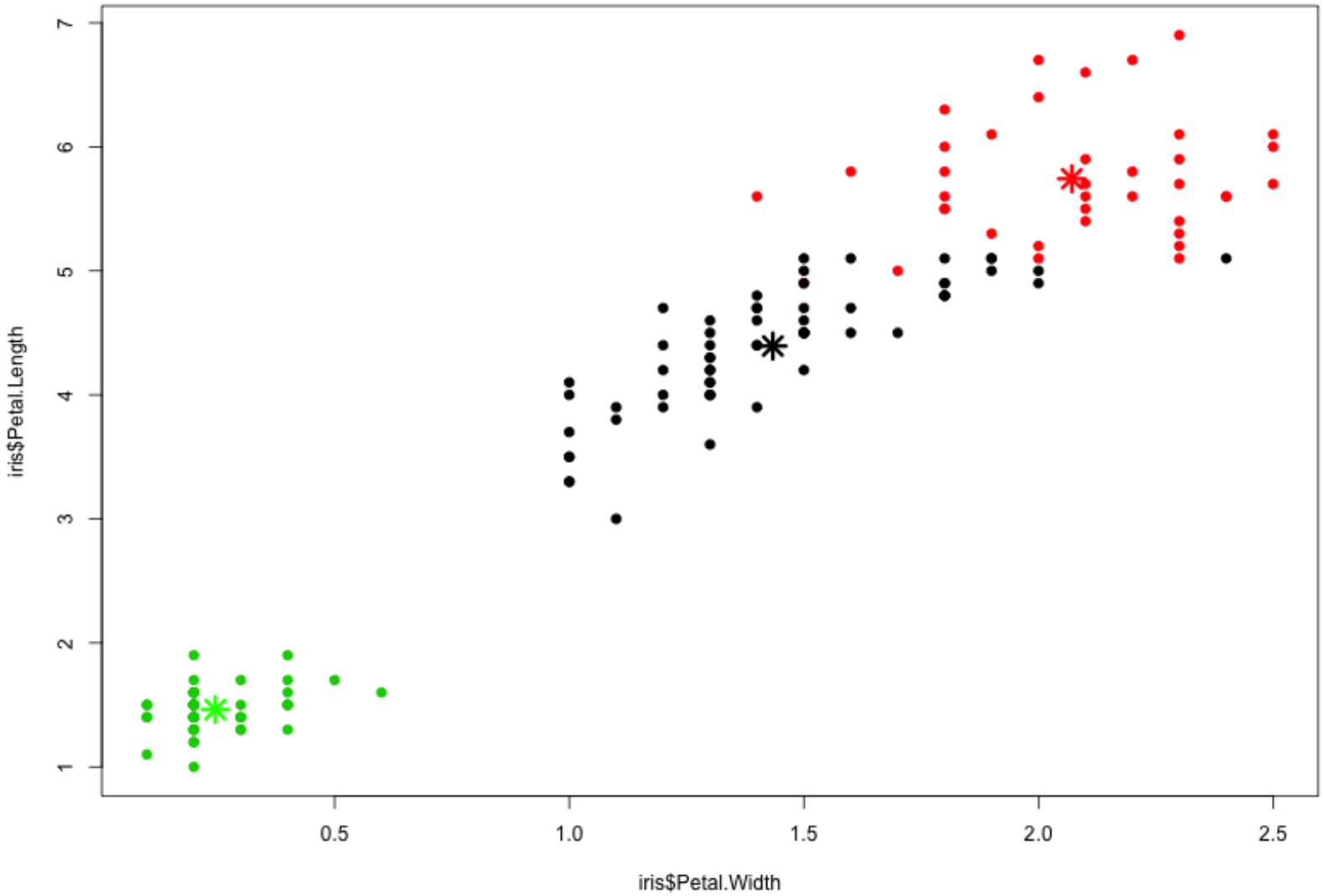
Será que as plantas deste gênero podem ser divididas em espécies?

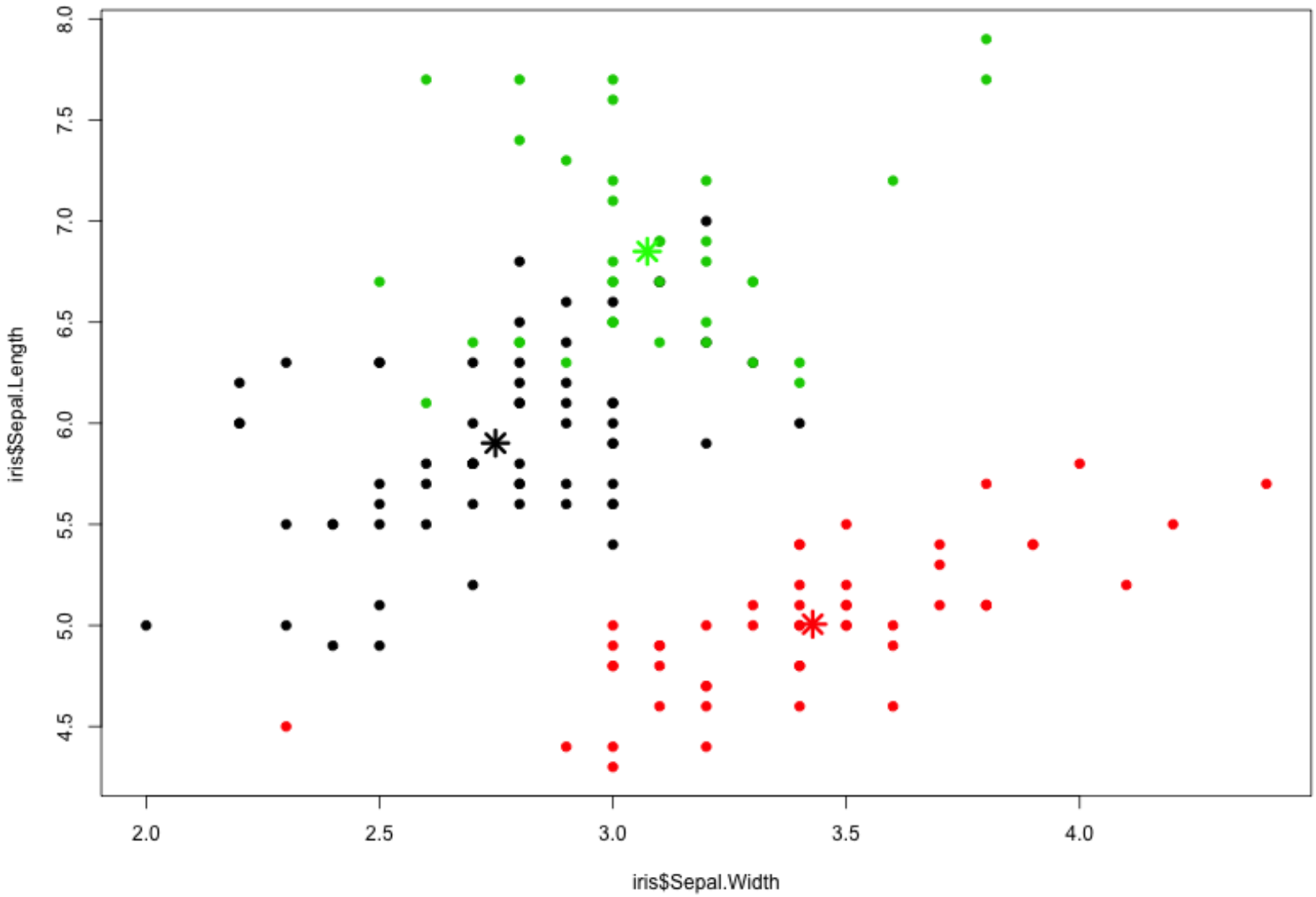
Aplicando o algoritmo K-means

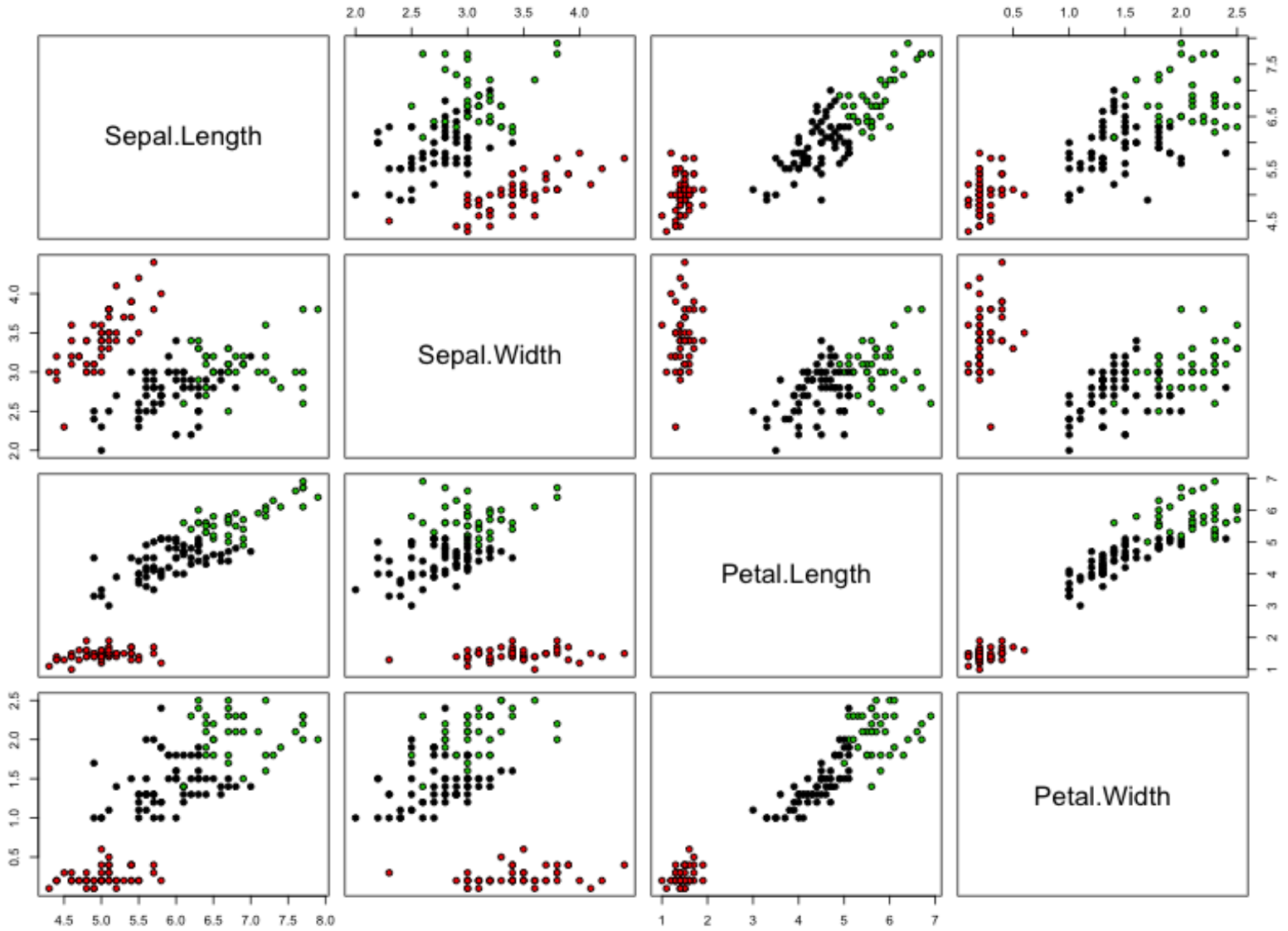
```
> model <- kmeans(iris, centers = 3)
> model
K-means clustering with 3 clusters of sizes 50, 62, 38

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.006000    3.428000     1.462000    0.246000
2     5.901613    2.748387     4.393548    1.433871
3     6.850000    3.073684     5.742105    2.071053

> model$withinss
[1] 15.15100 39.82097 23.87947
```

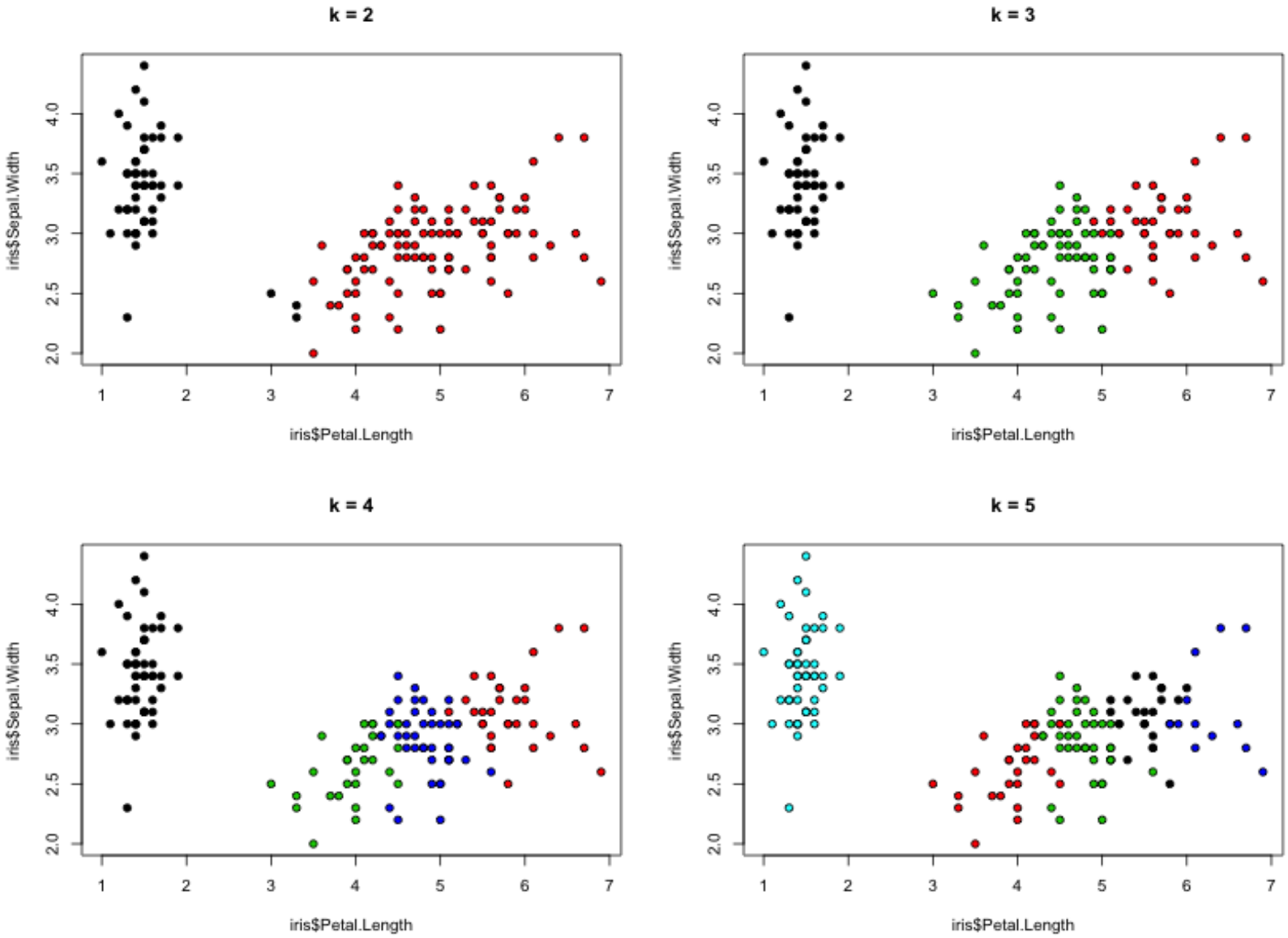

iris$Petal.Length
iris$Petal.Width
1
2
3
4
5
6
7
0.5
1.0
1.5
2.0
2.5


iris$Sepal.Length
iris$Sepal.Width
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
2.0
2.5
3.0
3.5
4.0

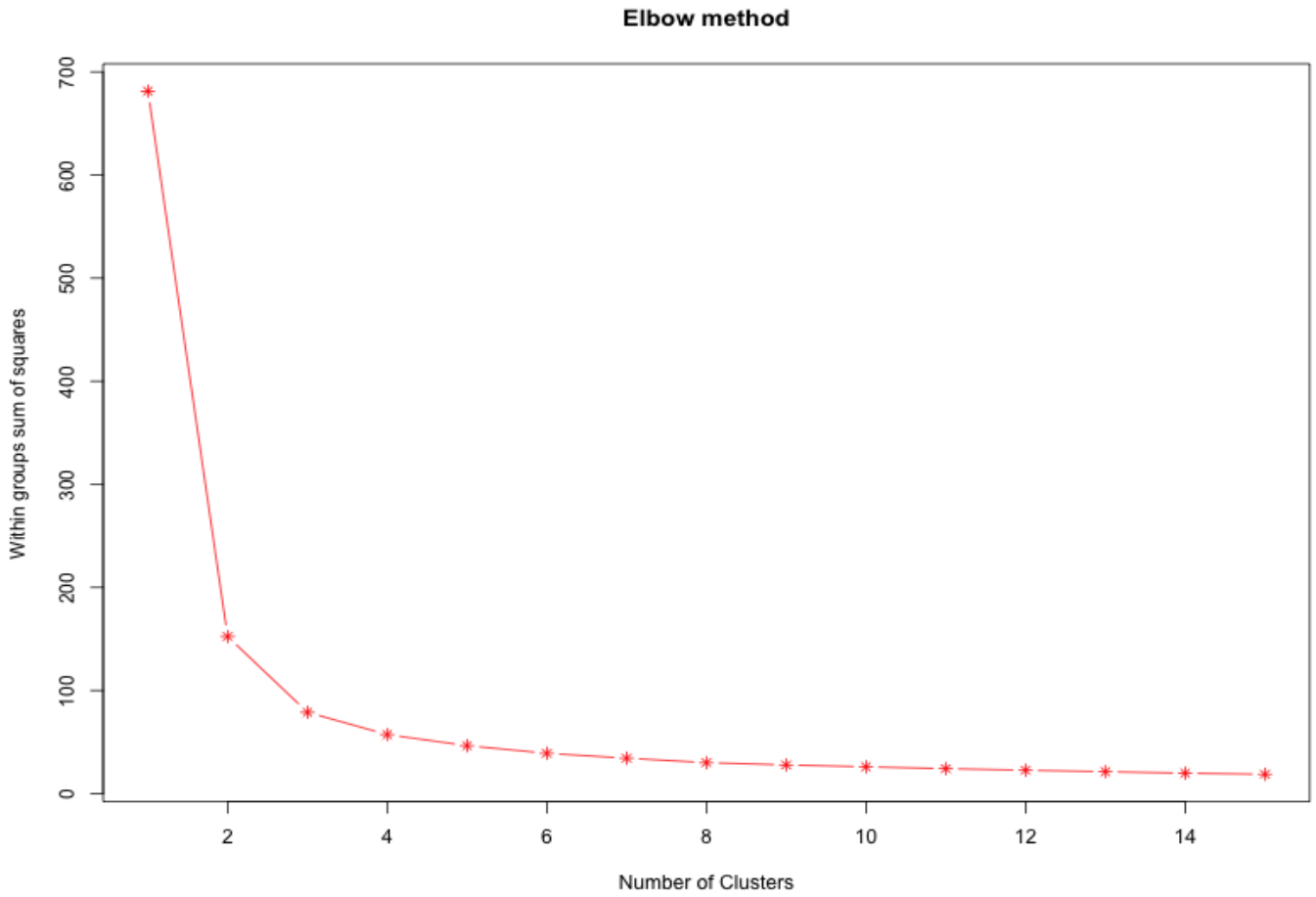

Sepal.Length
Sepal.Width
Petal.Length
Petal.Width

Dúvida...

Qual é o melhor número de clusters (k)?


k = 2
k = 3
k = 4
k = 5
iris$Sepal.Width
iris$Petal.Length
2.0
2.5
3.0
3.5
4.0
1
2
3
4
5
6
7

Como determinar o melhor k?



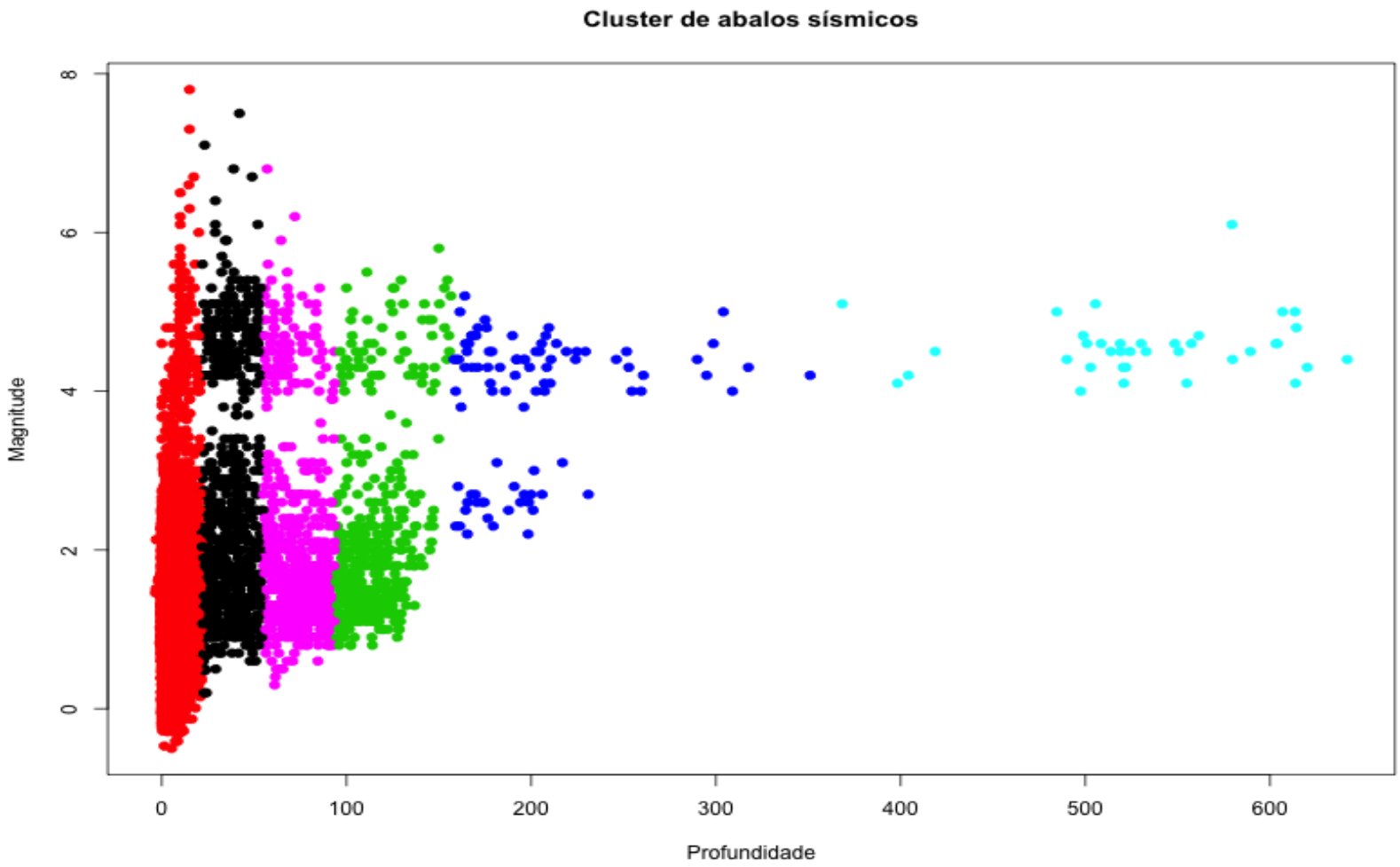
A medida de distribuição dos pontos normalmente empregada é *sum of squared errors*.

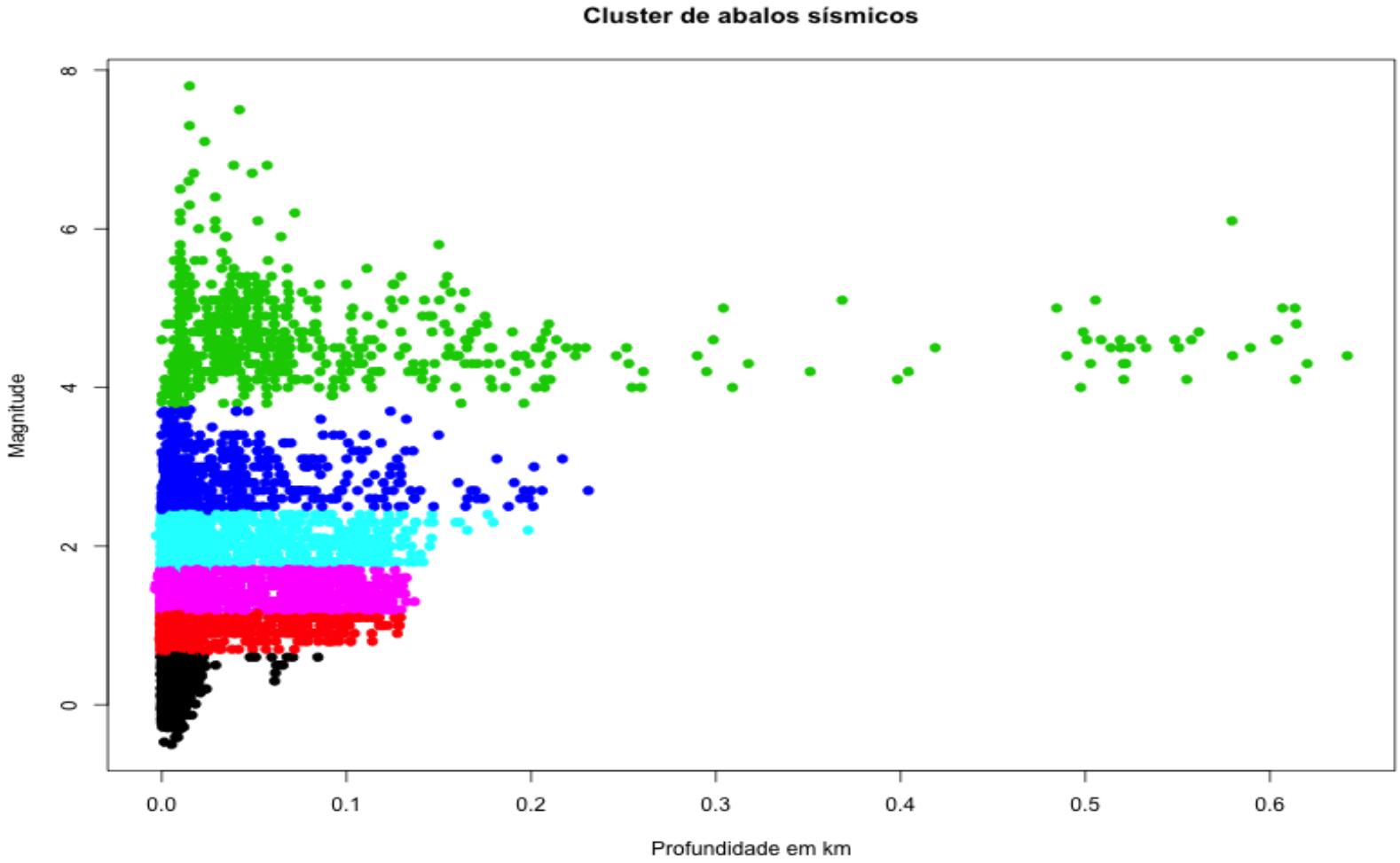
Exercícios

- Usando o dataset **survey** da biblioteca **UsingR**, identifique clusters de pessoas com base apenas nos atributos **Wr.Hnd** e **NW.Hnd**.
- Fazendo uso dos dados coletados em [a], agrupe os abalos sísmicos levando-se em consideração a magnitude e profundidade do abalo.

[a] http://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_month.csv

Clusters com dados em escalas diferentes




Cluster de abalos sísmicos
Magnitude
0
2
4
6
8
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Profundidade em km

Rescaling data

Feature scaling:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3}$$

Standardization:

$$x_{new} = \frac{x - mean(x)}{x_{max} - x_{min}} \tag{4}$$

$$x_{new} = \frac{x - \mu}{\sigma} = \frac{x - mean(x)}{sd(x)} \tag{5}$$

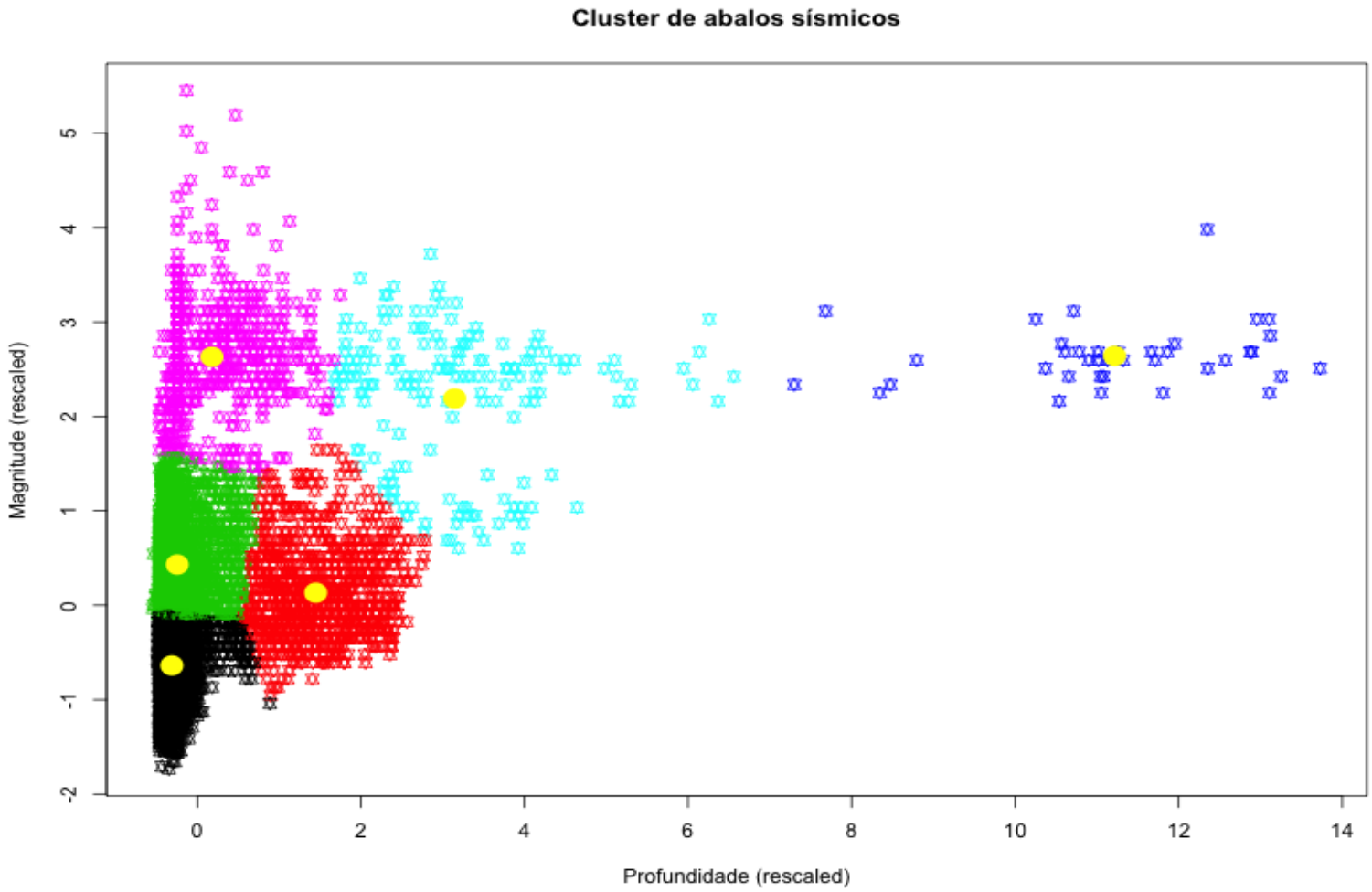
Cluster com dados normalizados

```
standardization <- function(x){
  return ((x - mean(x)) / sd(x))
}

filtrados$depthR <- standardization(filtrados$depth)
filtrados$magR <- standardization(filtrados$mag)
elbow(filtrados[,c('depthR','magR')])
model <- kmeans(filtrados[,c('depthR','magR')], centers = 6)

plot(filtrados$depthR, filtrados$magR,
     col=model$cluster,
     pch=11, main="Cluster de abalos sísmicos",
     xlab="Profundidade (rescaled)", ylab="Magnitude (rescaled)")
points(model$centers, col = "yellow", pch=19,cex=2,lwd=3)
```

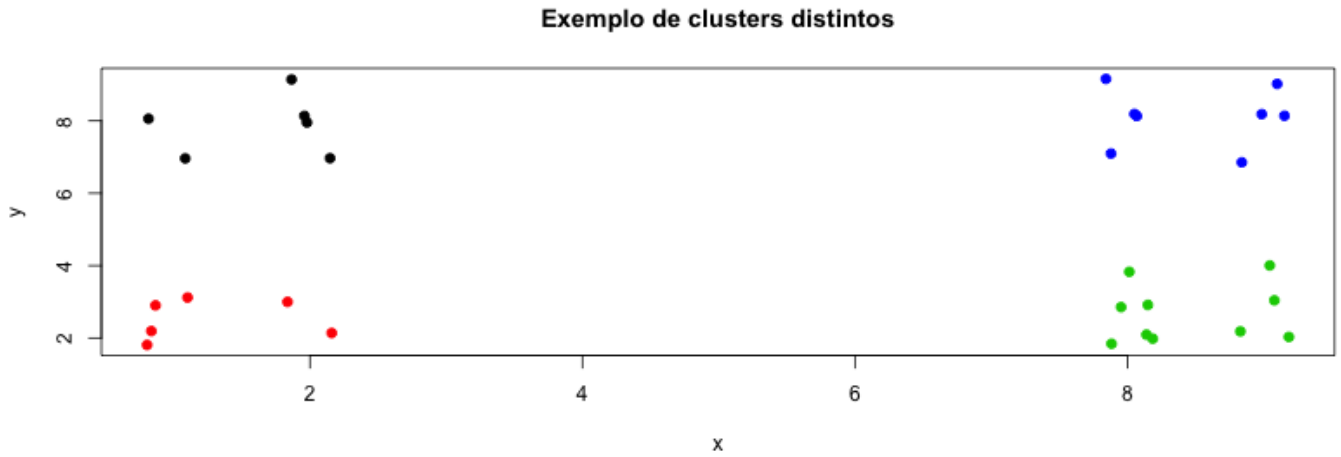
Cluster com dados normalizados

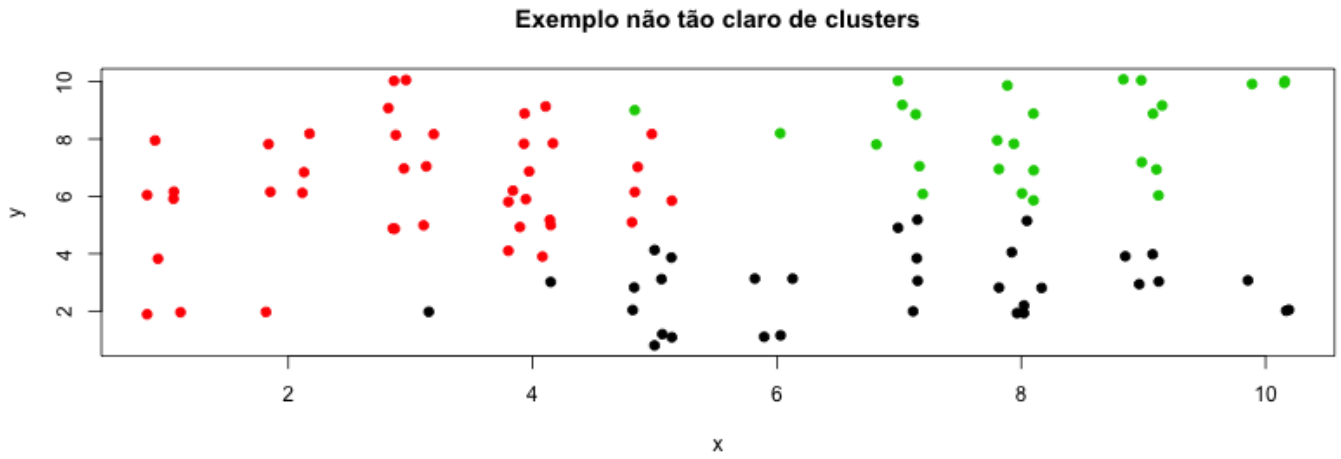


Diagnóstico

- Os clusters estão bem separados uns dos outros?
- Os centroides dos clusters estão bem separados uns dos outros?
- Existe algum cluster com poucos pontos?
- Os pontos de cada cluster estão bem agrupados?

Diagnóstico





Trabalhando com dados qualitativos

- O algoritmo k-means trabalha apenas com dados numéricos, pois utiliza a distância euclidiana como função para calcular a distância entre objetos.
- Para trabalhar com dados qualitativos é necessário fazer uso de outra função de distância, por exemplo a **distância de Hamming**.

Distância de Hamming

$$d(x_i, x_j) = \sum_{q=1}^{d} \alpha(x_i^q, x_j^q) \tag{6}$$

$$\alpha(x_i^q, x_j^q) = \begin{cases} 1 & \text{if } x_i^q \neq x_j^q \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Cluster de valores categóricos no R

- Função **kmodes** do pacote **klaR**
- Exemplo na conta https://github.com/fbarth, projeto posGraducao, código ExemplosClustering/script/clusteringValoresCategoricos.R

Alguns cuidados

- Que atributos devem ser incluídos na análise?
- Que unidades de medida (por exemplo, milhas, kilômetros, metros) devem ser utilizados em cada atributo?
- Os atributos precisam ser normalizados?
- Que outras considerações devem ser aplicadas?

Considerações adicionais

- O algoritmo k-means é sensível com relação aos pontos iniciais escolhidos para os centroides.
- Por isso, é importante executar várias vezes o algoritmo k-means para o mesmo **K** e escolher o resultado de cluster com menor WSS (*Within sum of squares*).
- No R isto é feito com o parâmetro **nstart** da função **k-means**.

Algoritmos para Agrupamento Hierárquico

Algoritmos para agrupamento hierárquico

Os algoritmos que utilizam a abordagem de agrupamento hierárquico podem implementar abordagens:

- **bottom-up (agglomerative clustering)**
- **top-down (divisive clustering)**

Agrupamento hierárquico **bottom-up**

Entrada: um conjunto $x = \{x_1, \cdots, x_n\}$ de objetos

e uma função sim: $P(X) \times P(X) \rightarrow \Re$

for i:=1 até n **do**

$c_i := \{x_i\}$ {Inicia com um agrupamento para cada objeto}

end for

j := n + 1

while $|C| > 1$ **do**

$(c_{n1}, c_{n2}) := \arg\max_{c_u, c_v \in C \times C} sim(c_u, c_v)$ {Em cada passo, os dois agrupamentos mais similares são determinados}

$c_j := c_{n1} \cup c_{n2}$ {Unidos em um novo agrupamento}

$C := C \setminus \{c_{n1}, c_{n2}\} \cup \{c_j\}$ {Elimina-se os dois agrupamentos mais similares e adiciona-se o novo agrupamento ao conjunto de agrupamentos}

$j := j + 1$

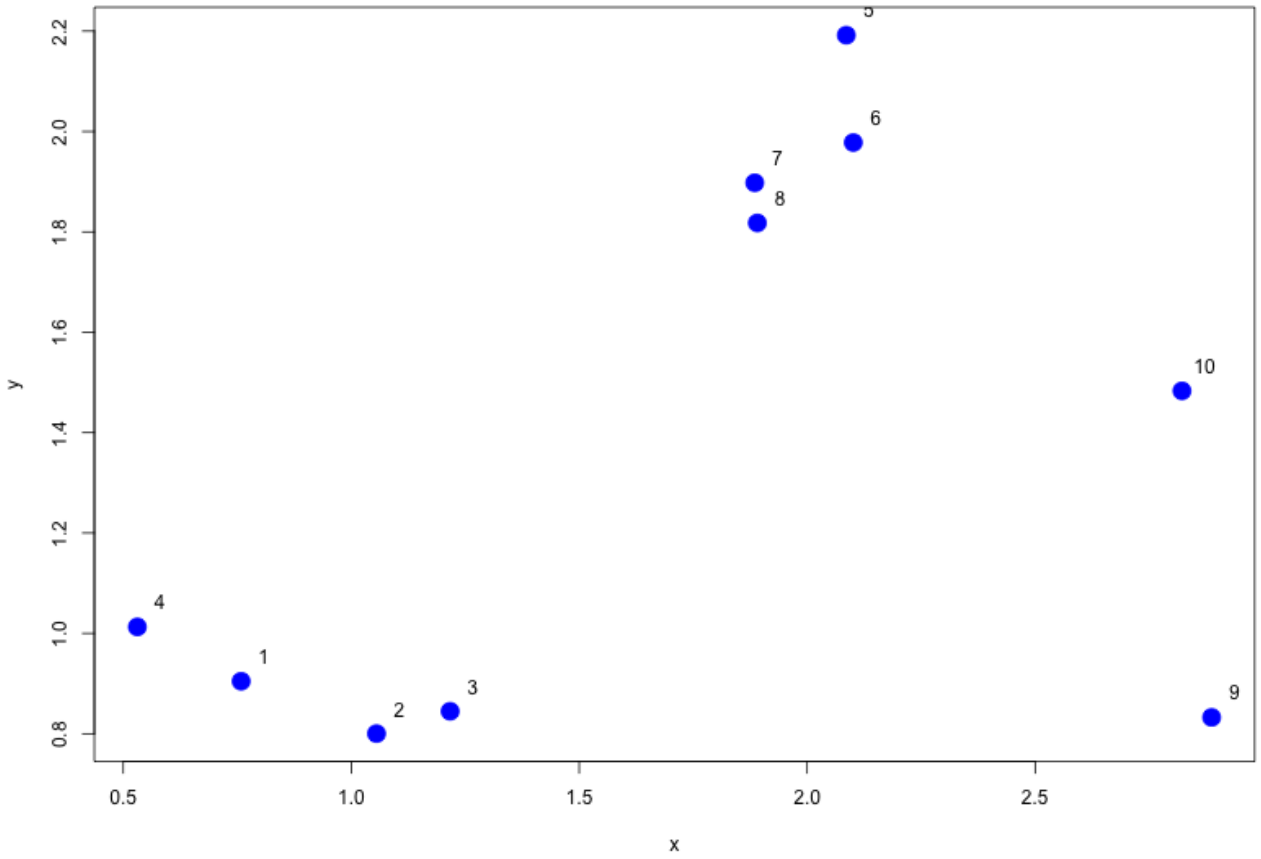
end while

Agrupamento hierárquico **bottom-up** - Função de similaridade

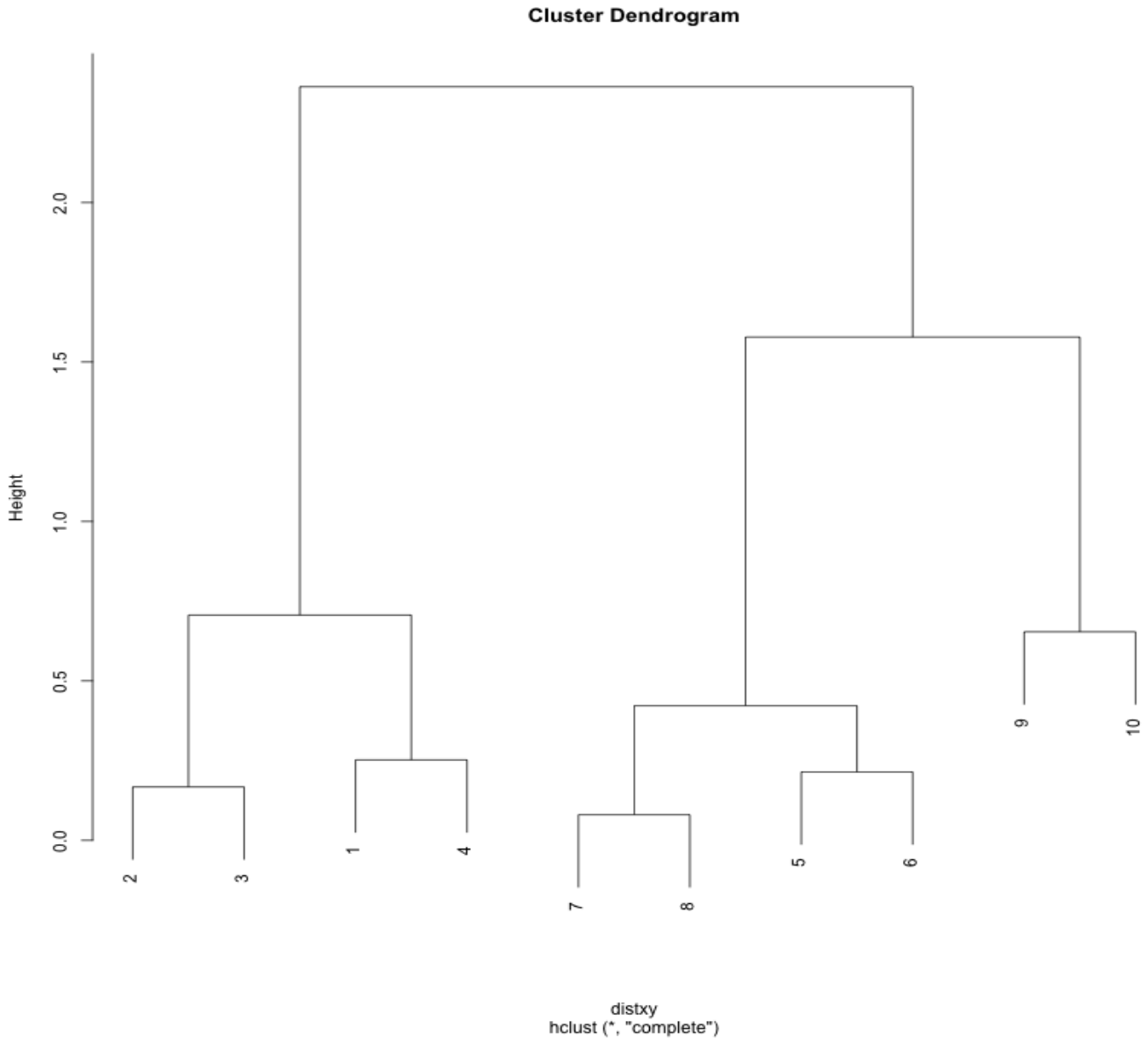
- A função de similaridade pode ser a distância Euclidiana:

$$| \overrightarrow{x} - \overrightarrow{y} | = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{8}$$

Funcionamento do algoritmo



	1	2	3	4	5	6	7	8	9	10
1	0.00	0.31	0.46	0.25	1.85	1.72	1.50	1.45	2.13	2.14
2	0.31	0.00	0.17	0.57	1.73	1.57	1.38	1.32	1.83	1.89
3	0.46	**0.17**	0.00	0.71	1.60	1.44	1.25	1.18	1.67	1.73
4	0.25	0.57	0.71	0.00	1.95	1.84	1.62	1.58	2.36	2.34
5	1.85	1.73	1.60	1.95	0.00	0.21	0.36	0.42	1.58	1.02
6	1.72	1.57	1.44	1.84	0.21	0.00	0.23	0.26	1.39	0.87
7	1.50	1.38	1.25	1.62	0.36	0.23	0.00	0.08	1.46	1.02
8	1.45	1.32	1.18	1.58	0.42	0.26	**0.08**	0.00	1.40	0.99
9	2.13	1.83	1.67	2.36	1.58	1.39	1.46	1.40	0.00	0.65
10	2.14	1.89	1.73	2.34	1.02	0.87	1.02	0.99	0.65	0.00


Cluster Dendrogram
Height
0.0
0.5
1.0
1.5
2.0
2
3
1
4
7
8
5
6
9
10
distxy
hclust (*, "complete")

Tipos de funções de similaridade

- ligação simples (**single link**): a similaridade entre dois agrupamentos é o melhor resultado retornado da similaridade entre os seus membros **mais** similares.
- ligação completa (**complete link**): a similaridade entre dois agrupamentos é o melhor resultado retornado da similaridade entre os seus membros **menos** similares.
- média do grupo (**group-average**): a similaridade entre dois agrupamentos é a **média** da similaridade entre os membros dos agrupamentos.

Agrupamento hierárquico **top-down**

Entrada: um conjunto $x = \{x_1, \cdots, x_n\}$ de objetos,
uma funcao de coesao coh: $P(X) \rightarrow \Re$
e uma funcao de divisao $split$: $P(X) \rightarrow P(X) \times P(X)$
$C := \{X\} (= \{c_1\})$ {Inicia com um agrupamento com todos os objetos}
$j := 1$
while $\{\exists c_i \in C \mid |c_i| > 1\}$ **do**
  $c_u := \arg\min_{c_v \in C} coh(c_v)$ {Determina que agrupamento eh o menos coerente}
  $(c_{j+1}, c_{j+2}) := split(c_u)$ {Divide o agrupamento}
  $C := C \setminus \{c_u\} \cup \{c_{j+1}, c_{j+2}\}$
  $j := j + 2$
end while

Restrição sobre os agrupamentos hierárquicos

Agrupamento hierárquico só faz sentido se a função de similaridade é monotônica decrescente das folhas para a raiz da árvore:

$$\forall c, c', c'' \subseteq S : \min(sim(c, c'), sim(c, c'')) \geq sim(c, c' \cup c'') \tag{9}$$

Considerações Finais

Algumas considerações sobre agrupamentos

- Um agrupamento é um grupo de objetos centrados em torno de um ponto central.
- Os agrupamentos mais compactos são os preferidos.

Sumário dos atributos dos algoritmos

Agrupamento hierárquico:

- É a melhor abordagem para análise exploratória de dados.
- Fornece mais informação que agrupamento plano.
- Menos eficiente que agrupamento plano (tempo e memória gastos).

Sumário dos atributos dos algoritmos

Agrupamento plano:

- É preferível se a eficiência é um atributo importante e se o conjunto de dados é muito grande.
- O algoritmo **K-means** é o método mais simples e deve ser usado sobre novos conjuntos de dados porque os resultados são geralmente suficientes.