Universidade Federal do Paraná

Laboratório de Estatística e Geoinformação - LEG

Aprendizado não supervisionado

Profs.: Eduardo Vargas Ferreira
Walmes Marques Zeviani

Métodos de aprendizagem não supervisionada

- Vamos discutir dois métodos:
  - ⋆ **Análise de Componentes Principais**: tenta explicar a estrutura de covariância através de combinações lineares de $X_1, X_2, \ldots, X_p$;
  - ⋆ **Clustering**: se trata de uma ampla classe de métodos para descobrir agrupamentos nos dados.
- Exemplos:
  - ⋆ Pacientes com câncer agrupados de acordo com similaridades nas expressões gênicas;
  - ⋆ Grupos de consumidores caracterizados pelos seus históricos de navegação e compras;
  - ⋆ Filmes agrupados pelas notas dos espectadores.

Análise de Componentes Principais (ACP)

- Suponha que temos $X_1, X_2, \ldots, X_p$. Então, temos p variáveis para reproduzir a variabilidade geral do sistema;
- Porém, é possível que parte dessa variabilidade seja explicada por um número mínimo $k < p$ de variáveis;
- **Componentes principais** são as sequências de combinações lineares de $X_1, X_2, \ldots, X_p$ que maximizam a sua variância;
- Genericamente, representa um novo sistema de coordenadas, rotacionando o original com $X_1, X_2, \ldots, X_p$ como coordenadas;
- Os novos eixos representam as direções de **máxima variabilidade**, de forma mais parcimoniosa para descrever a estrutura de covariância;

Análise de Componentes Principais (ACP)

- A **primeira componente principal** de um conjunto de características $X_1, X_2, \ldots, X_p$ é a combinação linear normalizada ($\sum_{j=1}^{p} \phi_{j1}^2 = 1$)

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \ldots + \phi_{p1} X_p,$$

em que $\phi_1 = [\phi_{11}, \ldots, \phi_{p1}]^t$ são as cargas da 1ª c.p.;

- Elas maximizam a $Var(Z_1) = \phi_1^t \Sigma \phi_1$, em que Σ é a matriz de covariância de $\boldsymbol{X} = [X_1, \ldots, X_p]^t$;

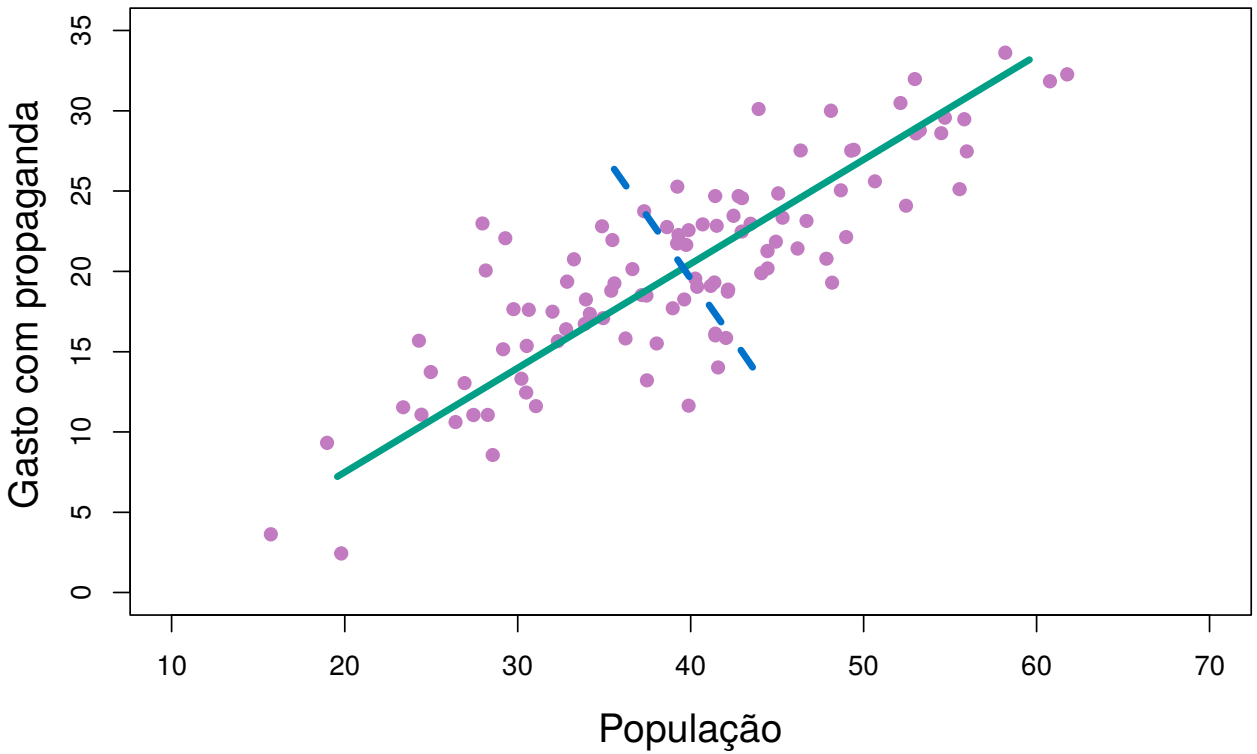
- A normalização deve-se ao fato da $Var(Z_1)$ aumentar a medida que ϕ_1 aumenta, assim eliminamos esta inconveniência;

- A **segunda componente principal** $\phi_2^t \boldsymbol{X}$ maximiza $Var(\phi_2^t \boldsymbol{X})$, sujeito a restrição $\phi_2^t \phi_2 = 1$ e $Cov(\phi_1^t \boldsymbol{X}, \phi_2^t \boldsymbol{X}) = 0$;

- A ***i*-ésima componente principal** $\phi_i^t \boldsymbol{X}$ maximiza $Var(\phi_i^t \boldsymbol{X})$, sujeito a restrição $\phi_i^t \phi_i = 1$ e $Cov(\phi_k^t \boldsymbol{X}, \phi_i^t \boldsymbol{X}) = 0, \forall k < i$.

ACP: exemplo

- O gráfico abaixo retrata o tamanho da população versus gasto com publicidade em 100 diferentes cidades;

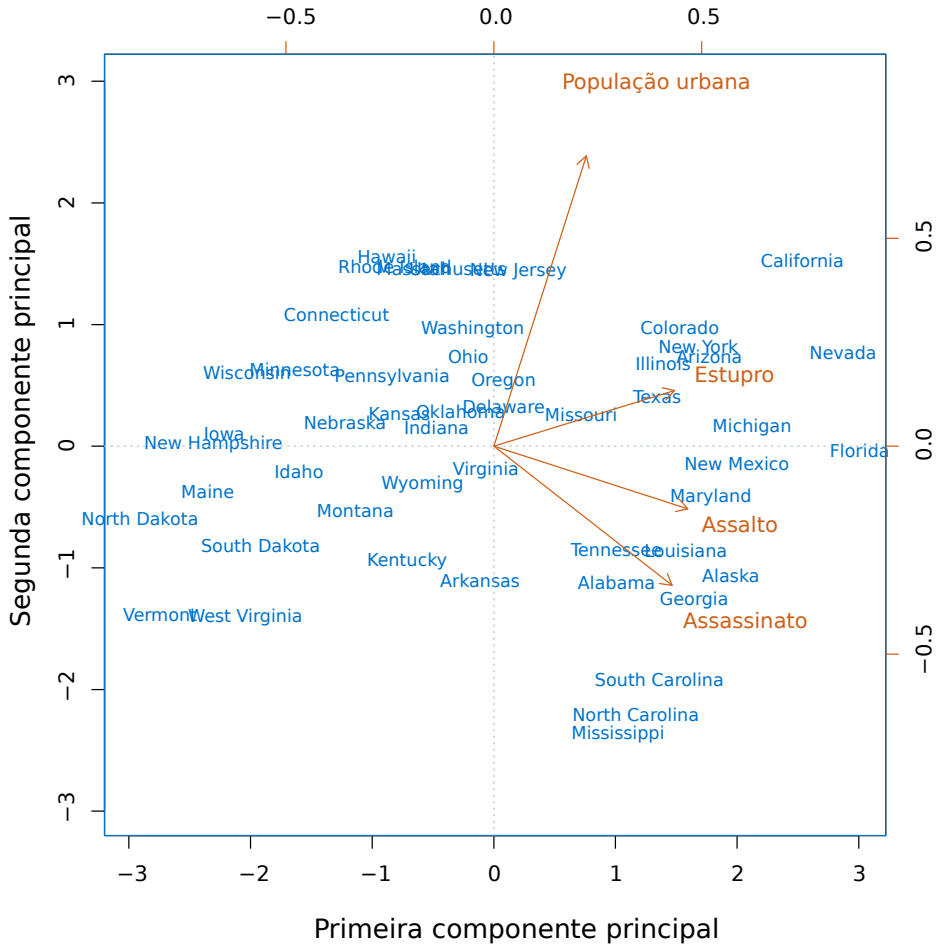


Exemplo: `USAarrests` data

- Os dados contém o número de prisões por 100.000 residentes nos 50 estados dos Estados Unidos;
- As prisões decorrem de: `Assalto`, `Assassinato` ou `Estupro`;
- Registrou-se, também, o percentual da população vivendo em área urbana de cada estado (`População urbana`);
- Dessa forma, temos $n = 50$ e $p = 4$.
- No gráfico a seguir são apresentadas as duas primeiras componentes principais para o dados em questão.

Exemplo: `USAarrests` data

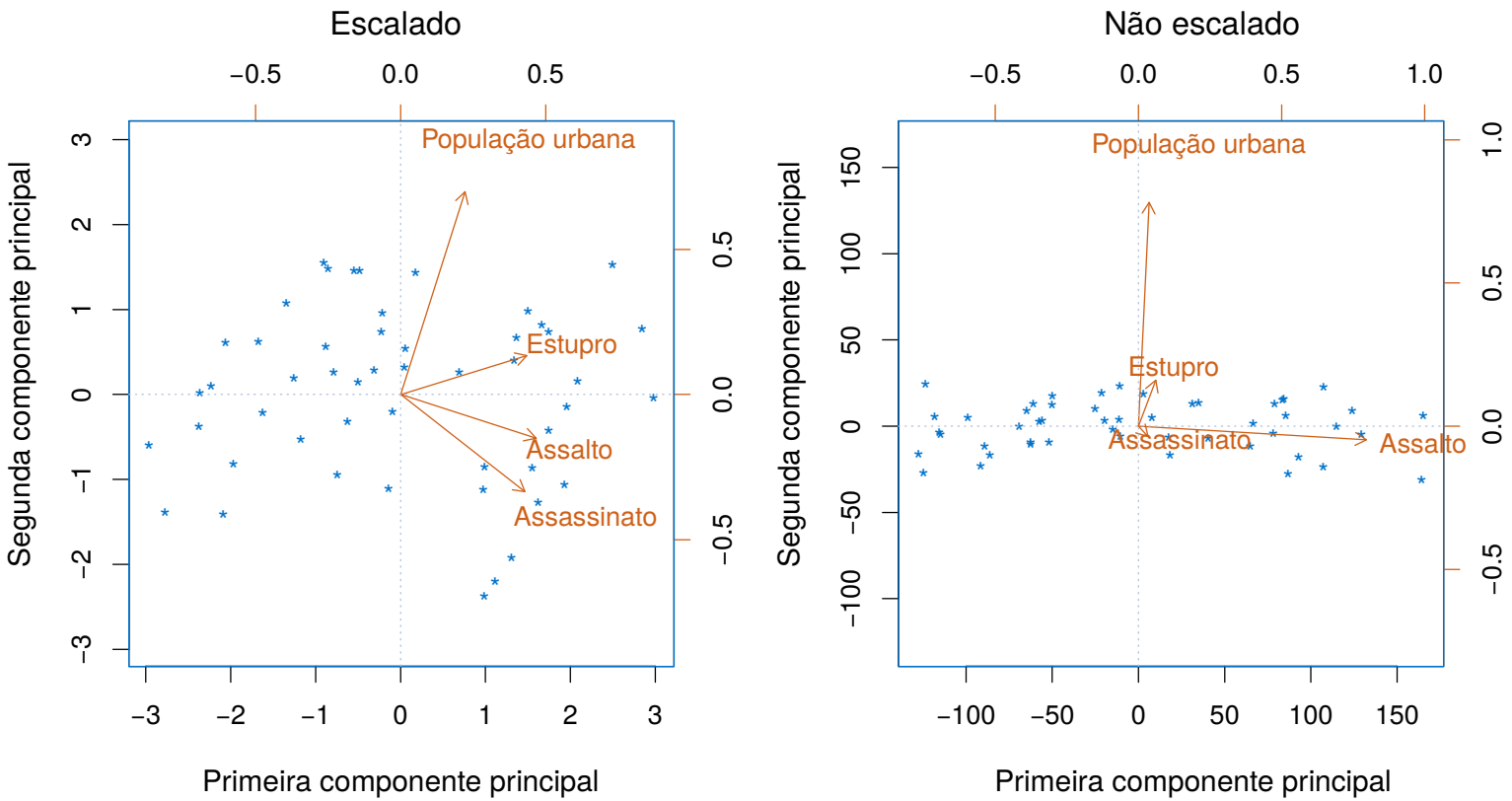
- Os dados contém o número de prisões por 100.000 residentes nos 50 estados dos Estados Unidos;



- Os nomes em azul representam o escore para as duas c.p.

Escalando as variáveis

- Se as variáveis estão em diferentes unidades é recomendável escalar cada uma para se ter um desvio padrão igual a 1.



Proporção da variância explicada

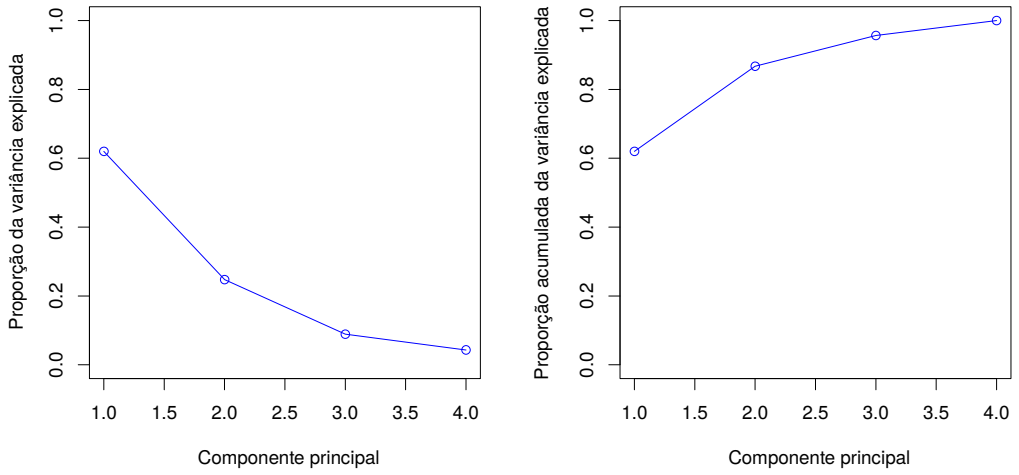
- A variância total presente nos dados é definida como

$$\sum_{j=1}^{p} Var(\boldsymbol{X}_j) = \lambda_1 + \lambda_2 + \ldots + \lambda_p = \sum_{j=1}^{p} Var(\boldsymbol{Z}_j)$$

- Assim, a proporção da variância explicada pela j-ésima componente principal é dada por

$$\frac{\lambda_j}{\lambda_1 + \lambda_2 + \ldots + \lambda_p}$$

- Abaixo, a proporção da variância explicada por cada uma das quatro c.p. referente ao `USAarrests` data;



Clustering

- **Clustering** refere-se ao conjunto de técnicas para encontrar subgrupos (ou *clusters*) a partir dos dados;
- Buscamos partições em grupos distintos, tal que observações dentro de cada grupo sejam similares entre si e diferentes dos demais;
- Veremos dois métodos:
  - ⋆ ***K*-means clustering**: procuramos partições dos dados em um número pré-determinado de clusters;
  - ⋆ **Hierarchical clustering**: não sabemos de antemão quantos clusters utilizaremos. Isso será feito através de uma representação visual;
  - ⋆ **DBSCAN**: um algoritmo de cluster baseado em densidades.

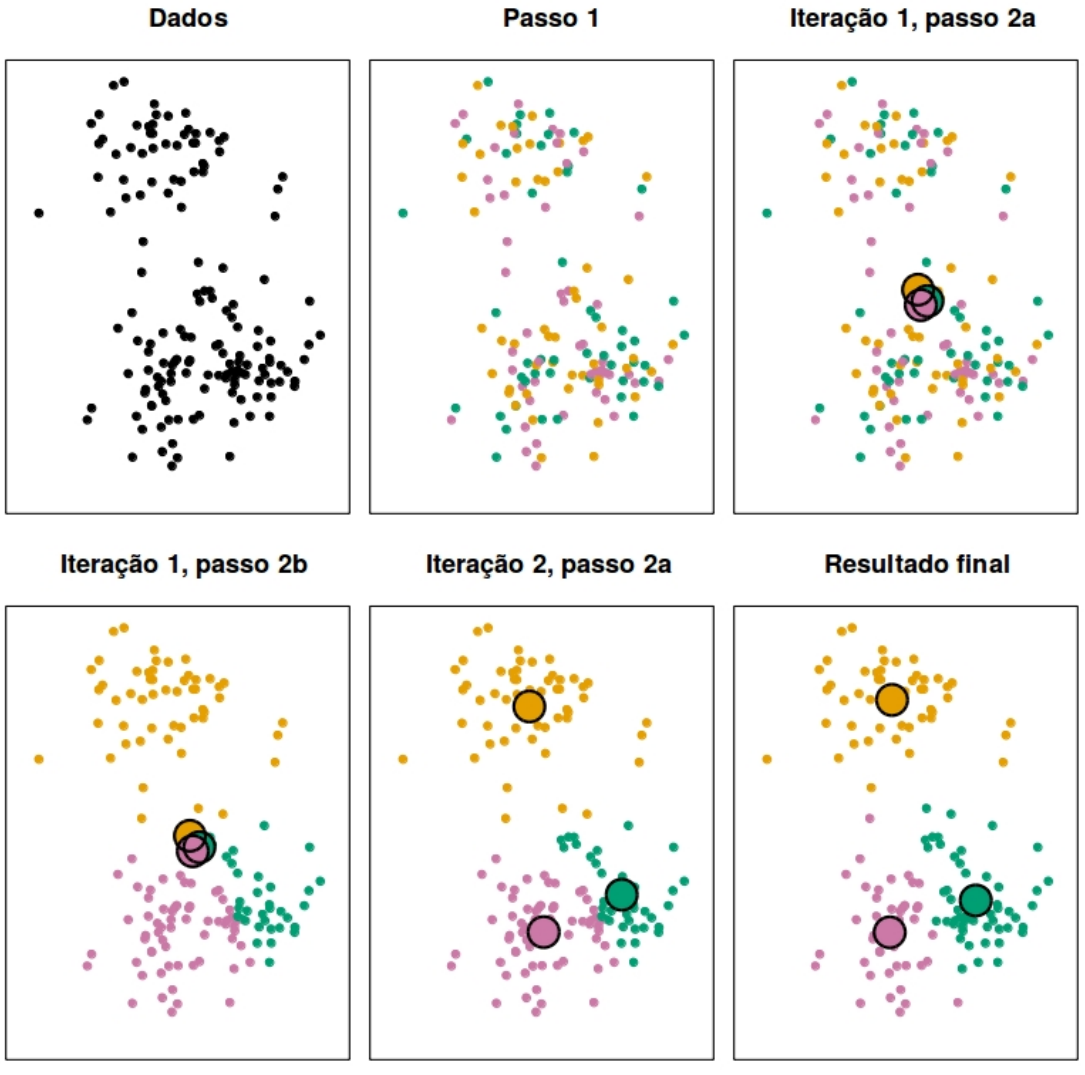
K-means

- A ideia do K-means é buscar agrupamentos, tal que a variação dentro de cada cluster seja tão pequena quanto possível;

Algoritmo

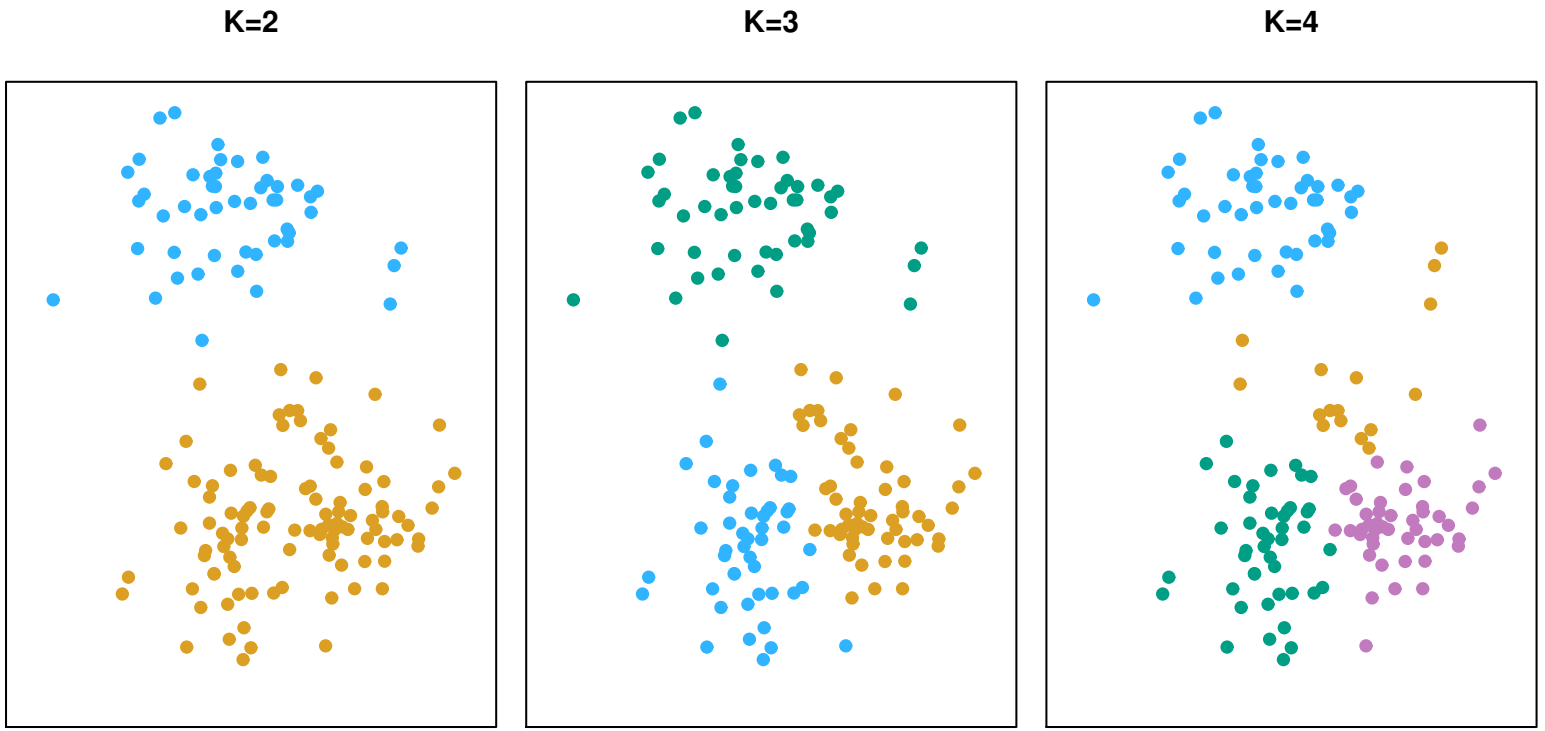
- ⋆ **Step 1:** Atribua, aleatoriamente, cada observação em um dos K clusters (este é o chute inicial);
- ⋆ **Step 2:** Itere até que os clusters se estabilizem:
  - (a) Para cada K cluster, calcule seu centroide;
  - (b) Atribua cada observação ao cluster mais próximo (menor distância Euclideana).

Detalhes do algoritmo



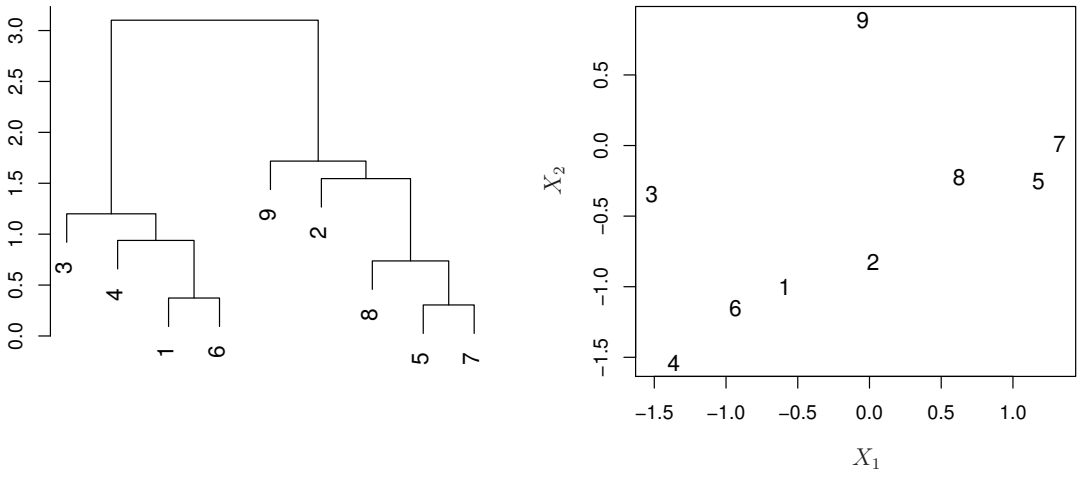
Exemplo simulado

- Os dados simulados consistem em 150 observações. Os painéis representam os resultados de K-means para diferentes K's;



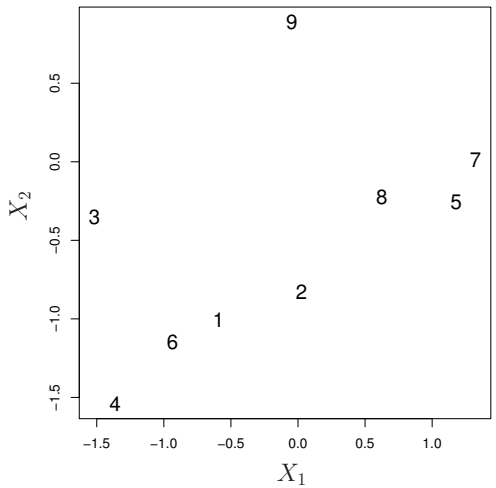
Hierarchical Clustering

- Como vimos, k-means exige que preespecifiquemos o número de clusters. O que pode ser uma desvantagem;
- **Hierarchical clustering** é uma abordagem alternativa, que não exige comprometimento com a escolha de K;



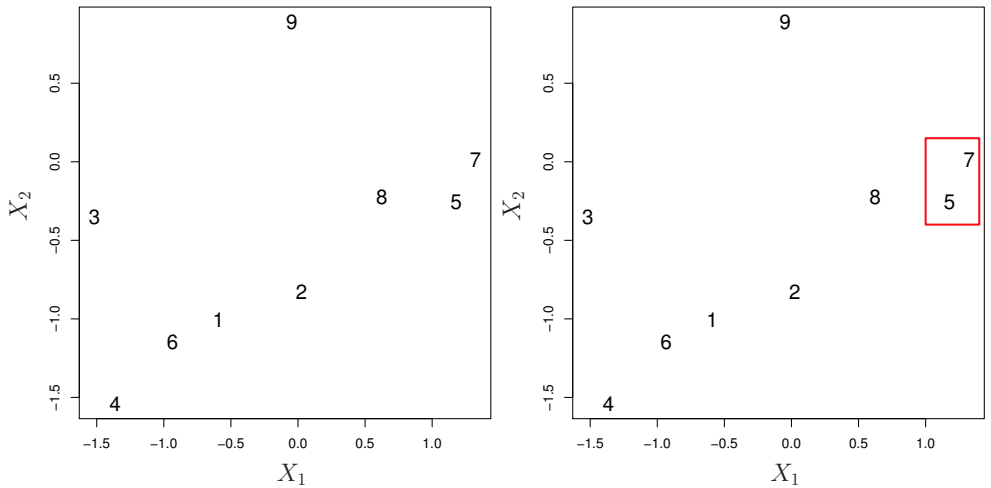
- A ideia é construir um dendrograma com folhas que se agrupam até chegar ao tronco;

Ideia do algoritmo



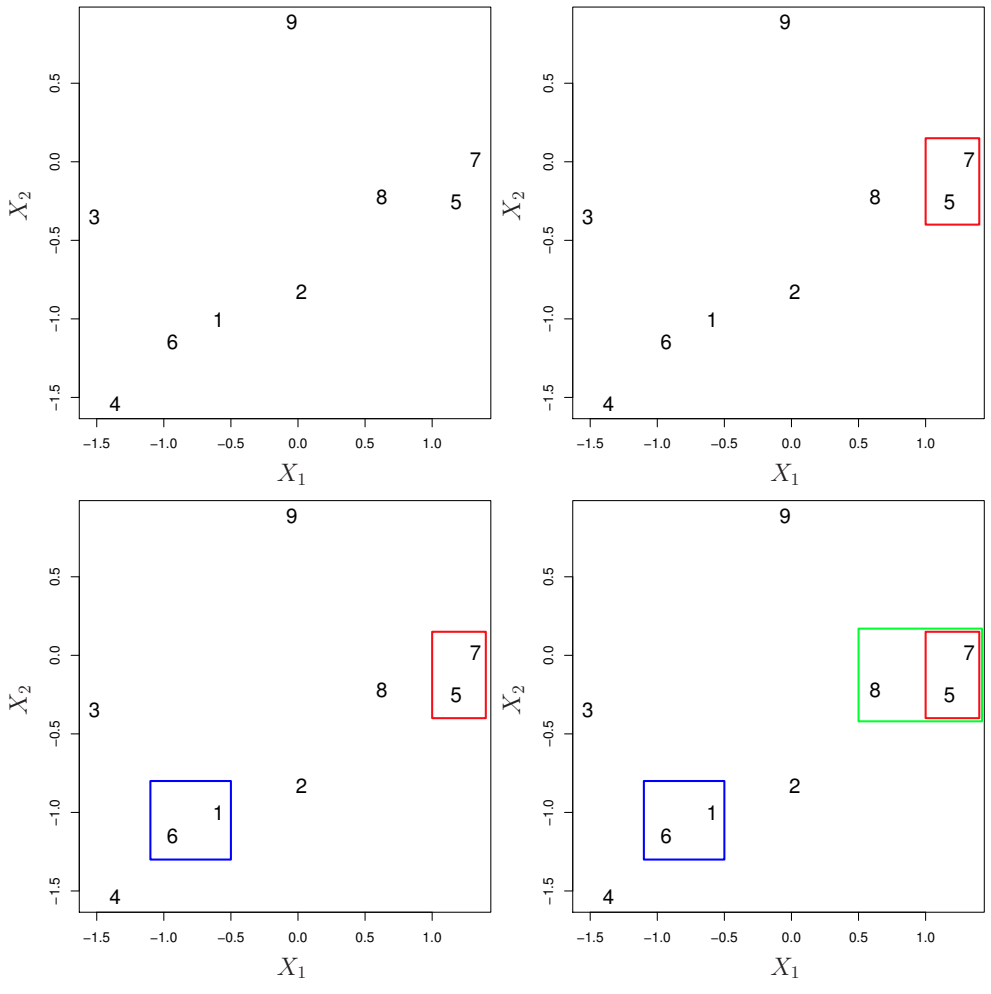
- Iniciamos com cada ponto sendo seu próprio cluster;
- Identificamos os dois clusters mais próximos e os agrupamos;
- Repetimos este processo até restar um cluster.

Ideia do algoritmo



- Iniciamos com cada ponto sendo seu próprio cluster;
- Identificamos os dois clusters mais próximos e os agrupamos;
- Repetimos este processo até restar um cluster.

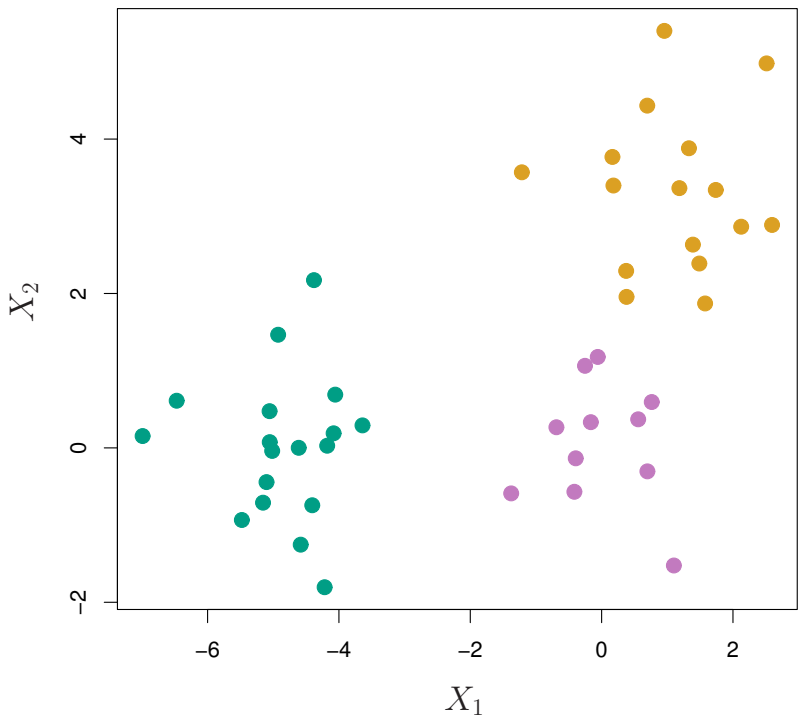
Ideia do algoritmo



- Iniciamos com cada ponto sendo seu próprio cluster;
- Identificamos os dois clusters mais próximos e os agrupamos;
- Repetimos este processo até restar um cluster.

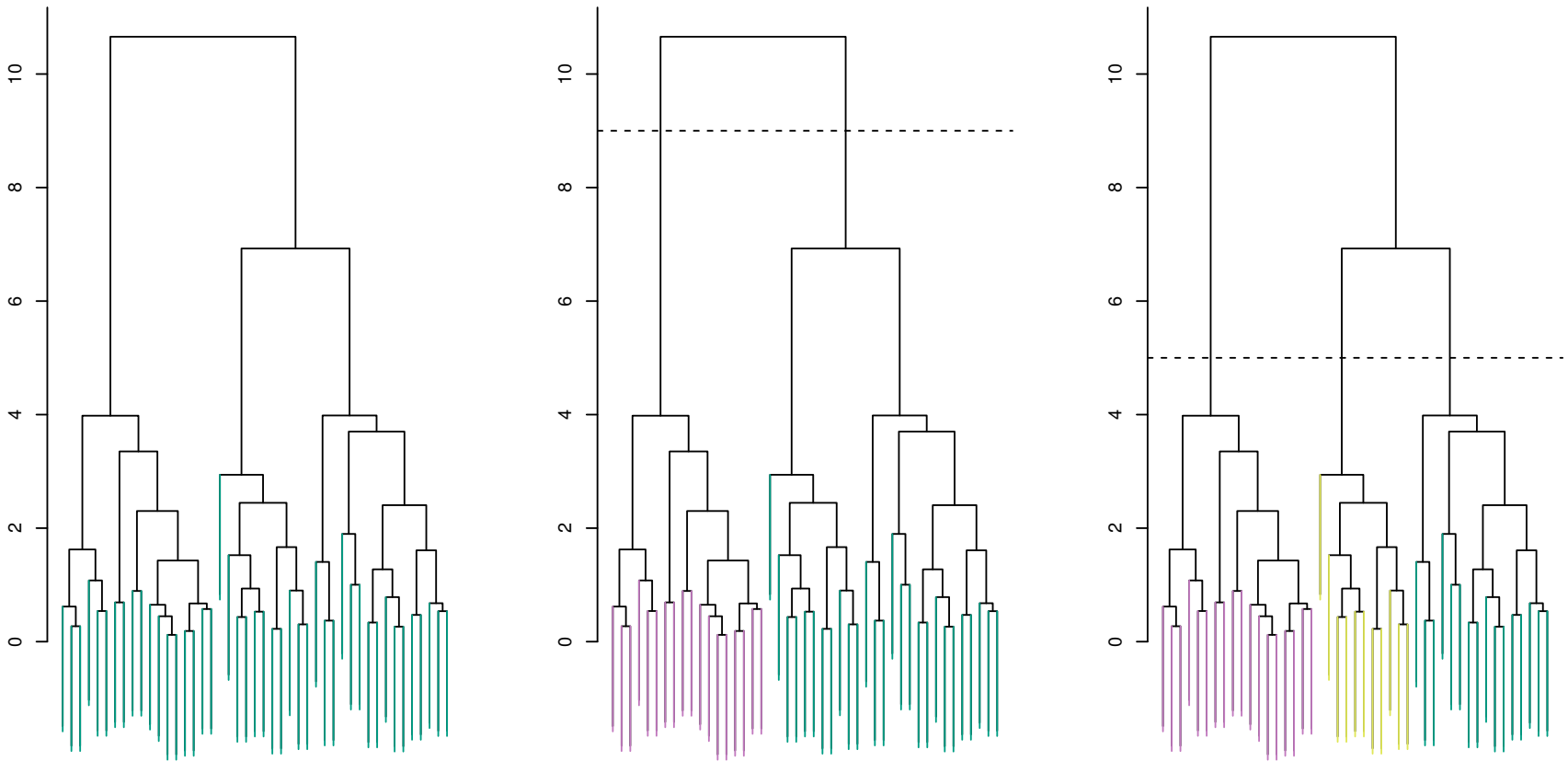
Exemplo

- Temos 45 observações, e 3 classes distintas (separadas por cores);



Exemplo

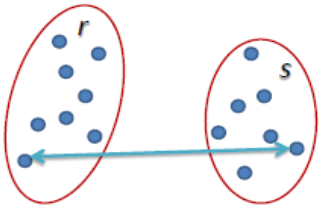
- Abaixo, três dendrogramas com diferentes alturas de corte (que resulta em clusters distintos);



Tipo de Linkage

Complete

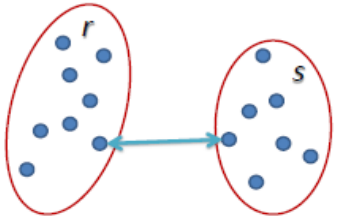
- Calculamos a máxima dissimilaridade entre os clusters.



$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

Single

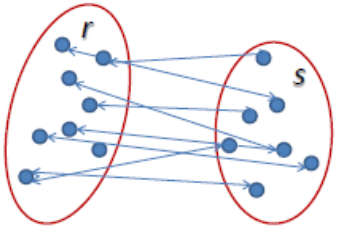
- Calculamos a mínima dissimilaridade entre os clusters.



$$L(r,s) = \min(D(x_{ri}, x_{sj}))$$

Average

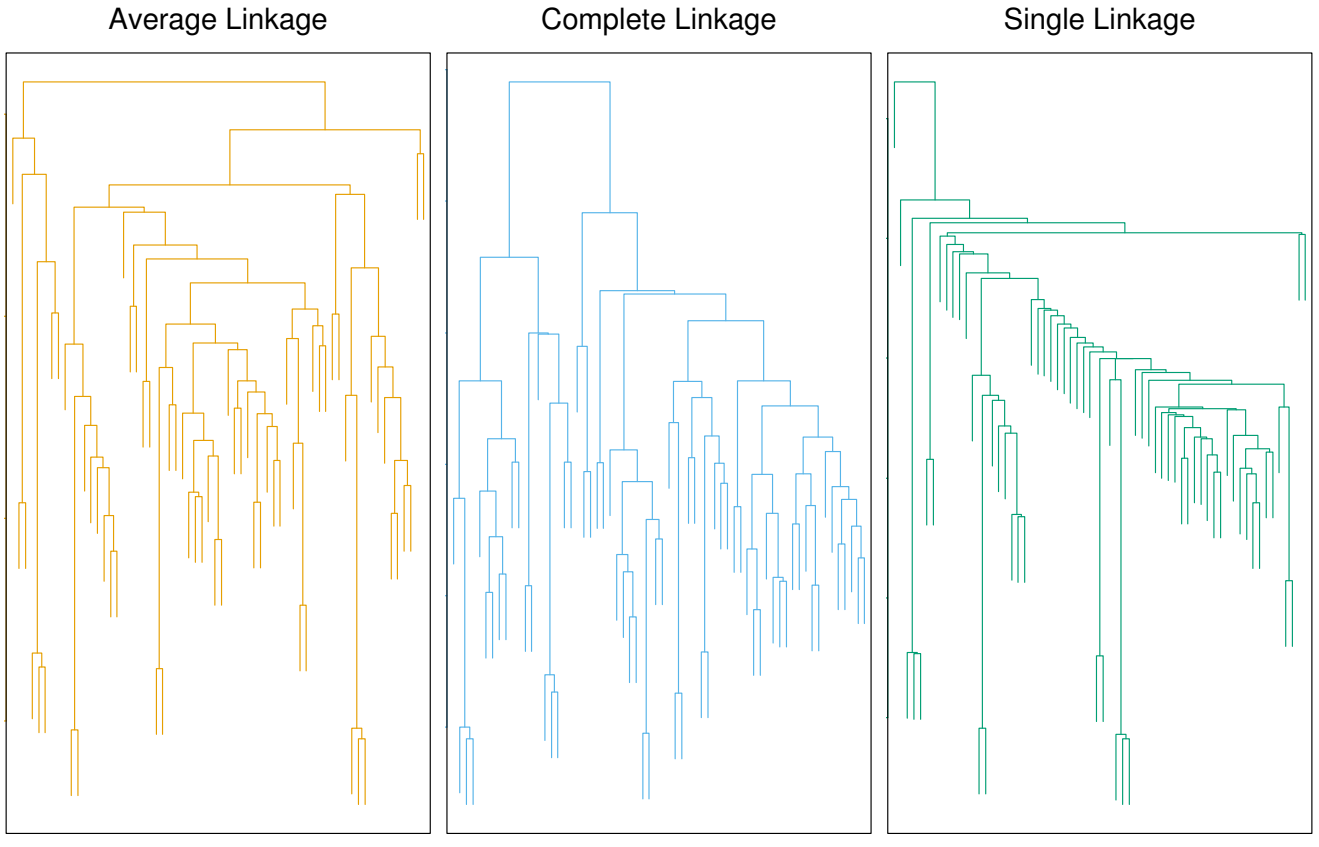
- Calculamos a dissimilaridade média entre os clusters.



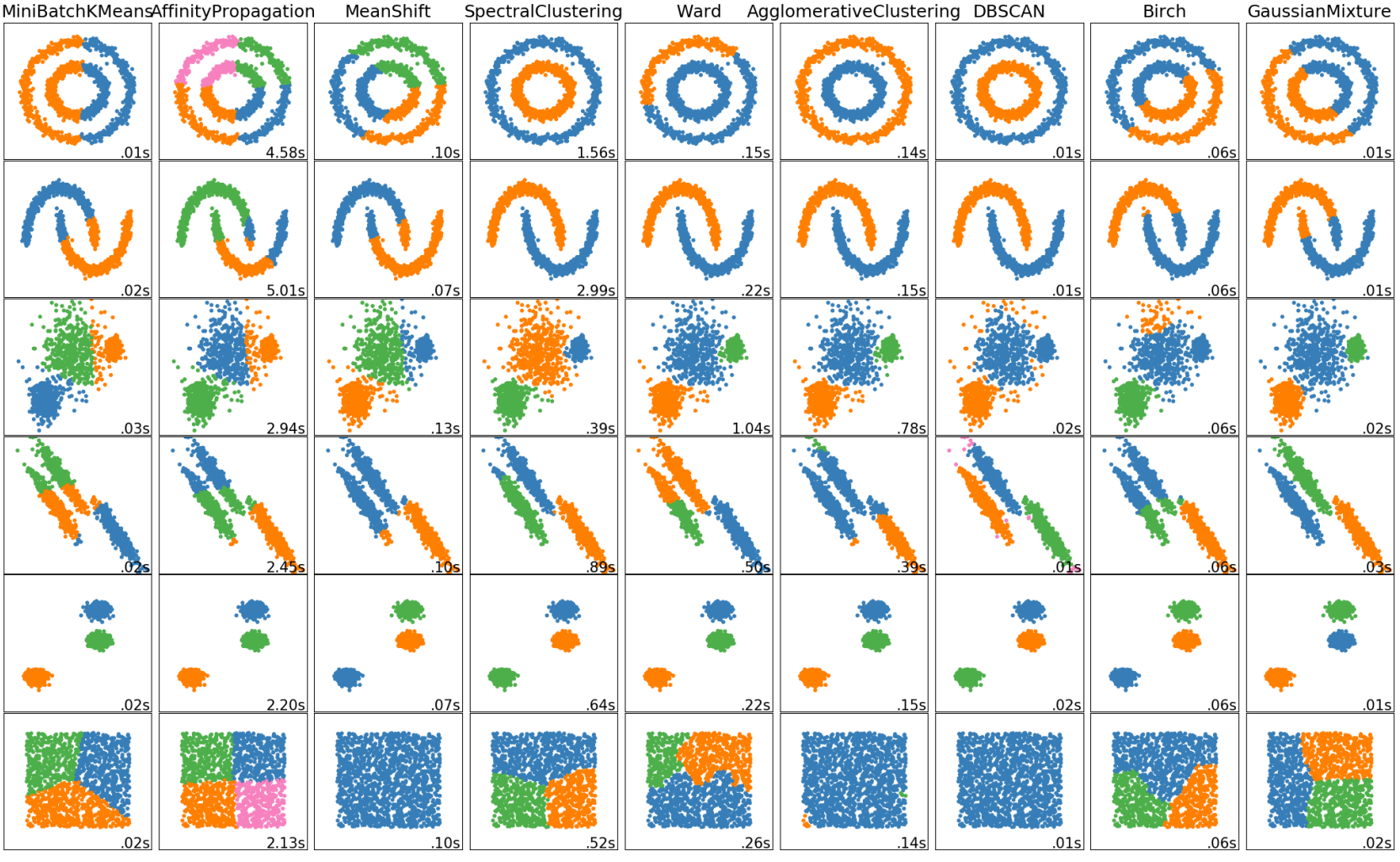
$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

Exemplo

- Em geral, **average** e **complete** linkage tendem a produzir agrupamentos mais equilibrados.

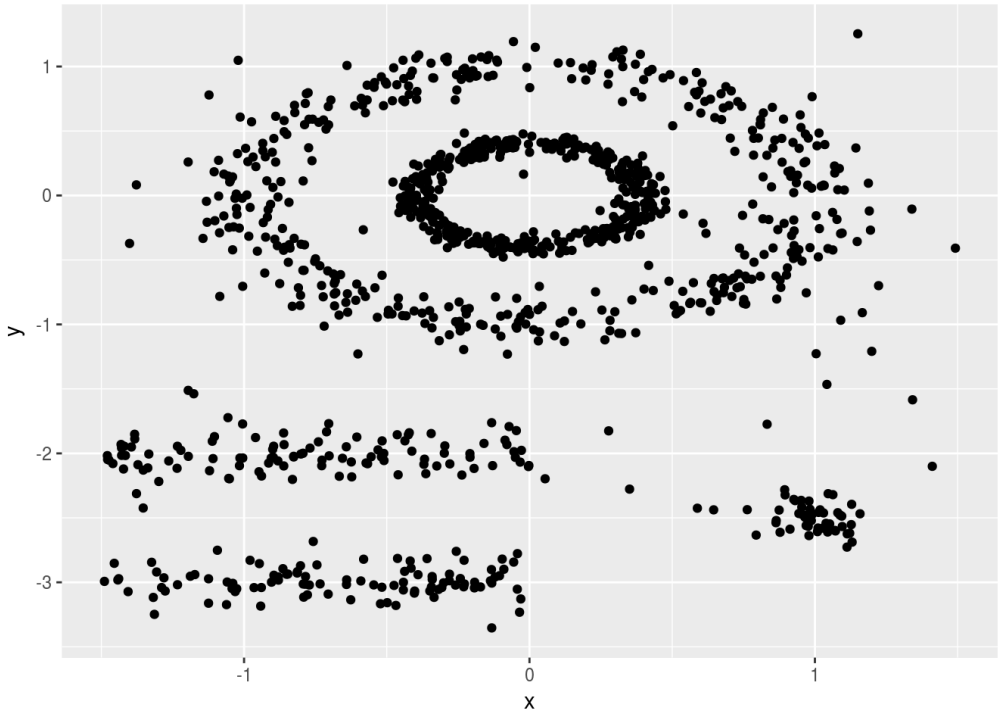


Outros tipos de clusters



DBSCAN

- Os métodos que vimos anteriormente são adequados para encontrar agrupamentos esféricos, em regiões bem definidas e ausentes de outliers.



- Entretanto, no mundo real, os clusters podem ter formas arbitrárias (oval, em forma de "S" etc.), e virem com outliers e ruídos.

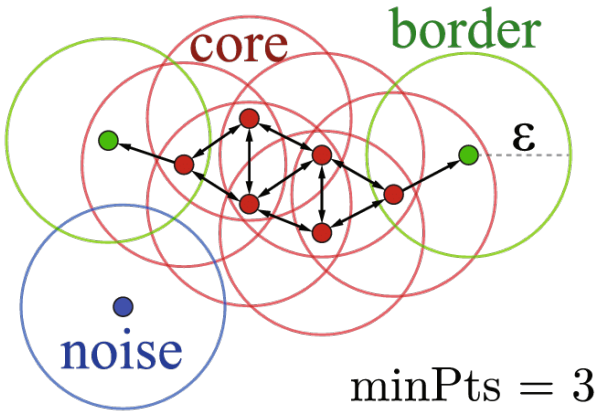
DBSCAN

- DBSCAN (*Density-Based Spatial Clustering and Application with Noise*) é um algoritmo de cluster baseado em densidade;
- A ideia básica da abordagem é derivada do método intuitivo humano de diferenciar regiões no espaço:

Clusters são regiões densas, separadas por regiões de menor densidade.

- Temos dois parâmetros de *tuning*:
  - ⋆ `eps`: que define o raio, ϵ, em torno do ponto x;
  - ⋆ `MinPts`: número mínimo de vizinhos dentro do raio ϵ.

Como o método funciona



- Qualquer ponto x, com uma quantidade de vizinhos maior ou igual a `MinPts` é considerado *core*;
- O ponto pertence a fronteira se o número de vizinhos $<$ `MinPts`, mas está contido no raio de algum *core*;
- Finalmente, se o ponto não é interior nem de fronteira, ele é considerado como ruído ou outlier.

Prós e contras do método

Vantagens

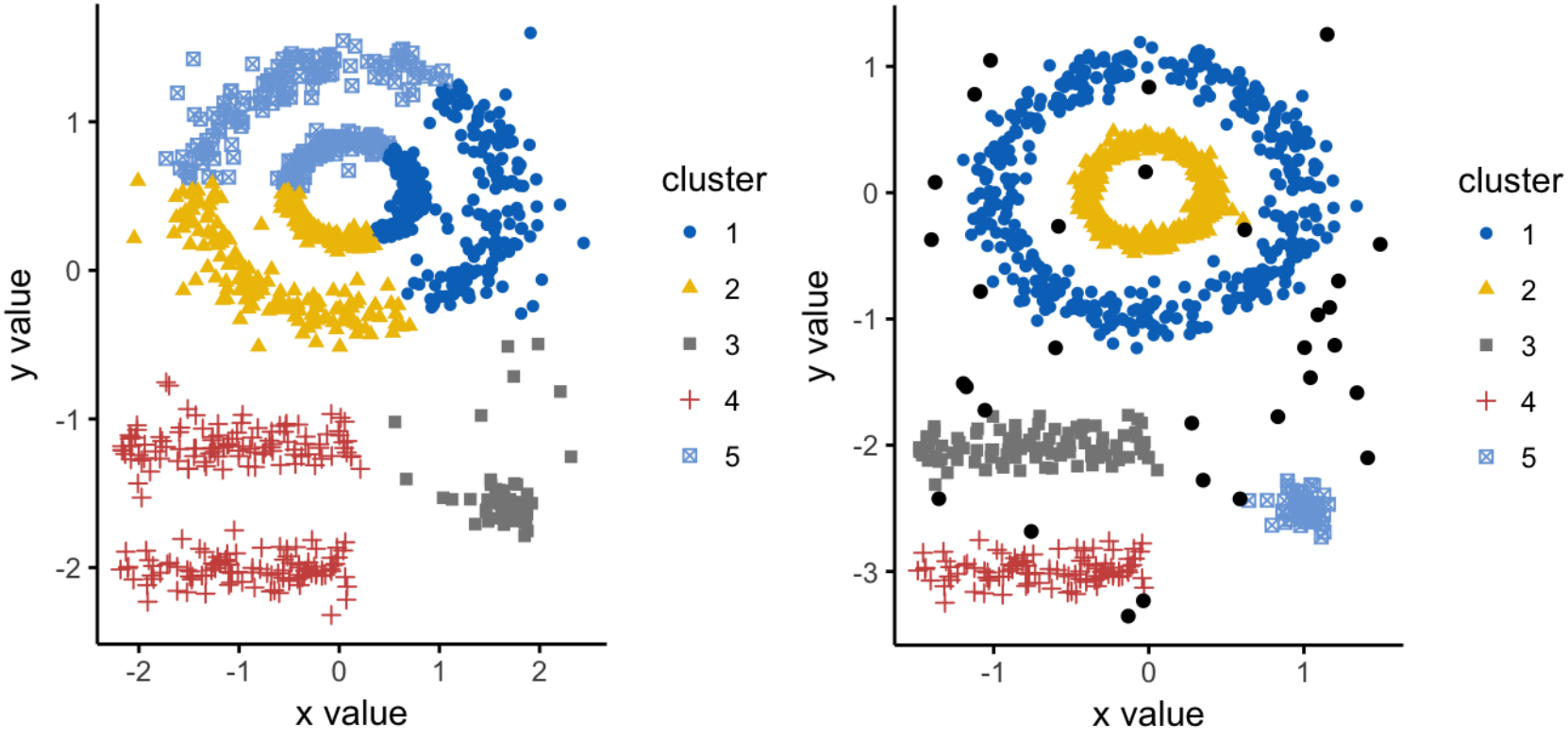
- Não requer um número predefinido de *clusters*;
- Podem ser de qualquer forma, incluindo não esféricos;
- A técnica é capaz de identificar dados de ruído (outliers).

Desvantagem

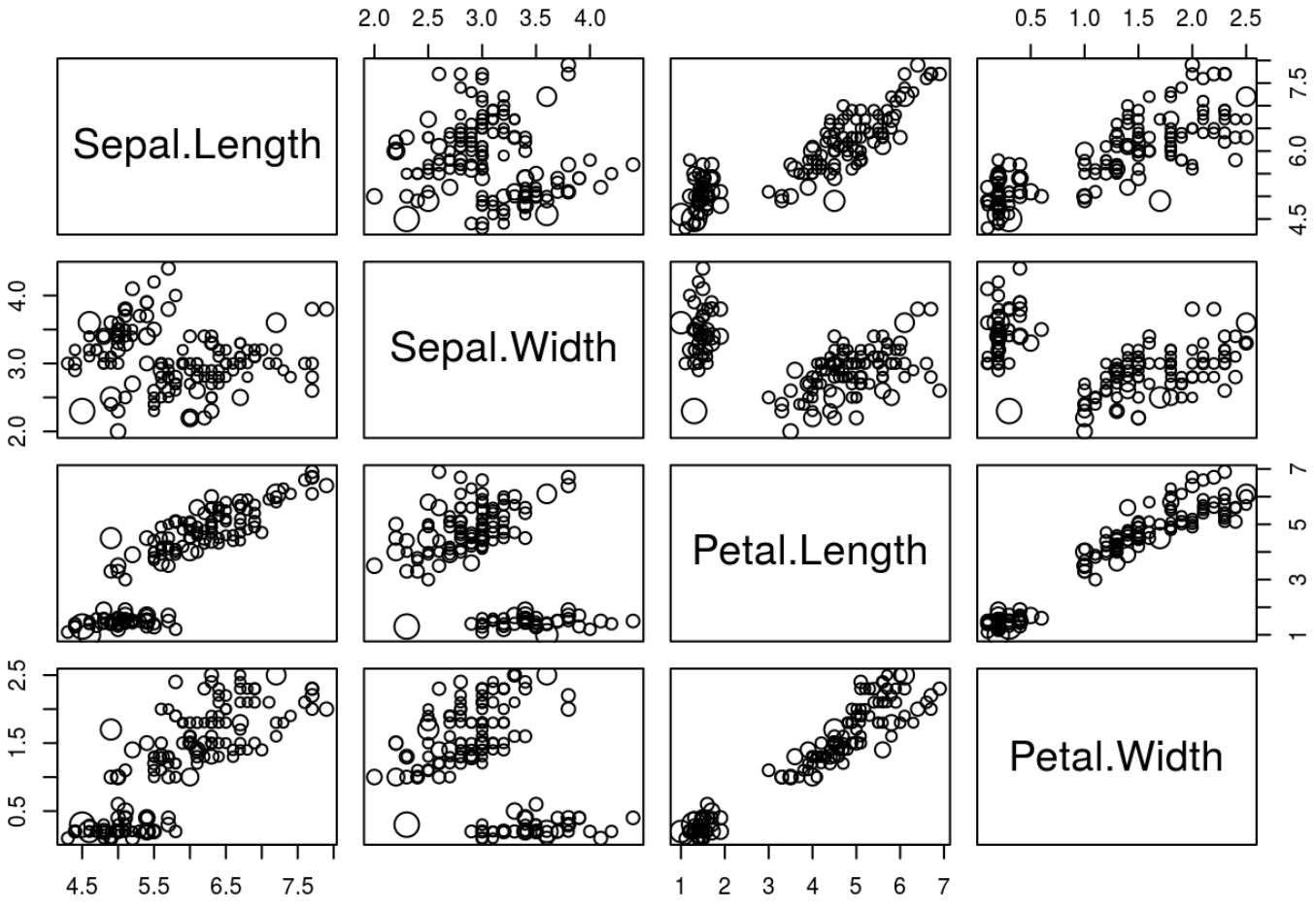
- Pode falhar se não houver queda de densidade entre clusters;
- É sensível aos parâmetros que definem a densidade de *tuning*;
- A configuração adequada pode exigir conhecimento e domínio.

K-means *vs* DBSCAN

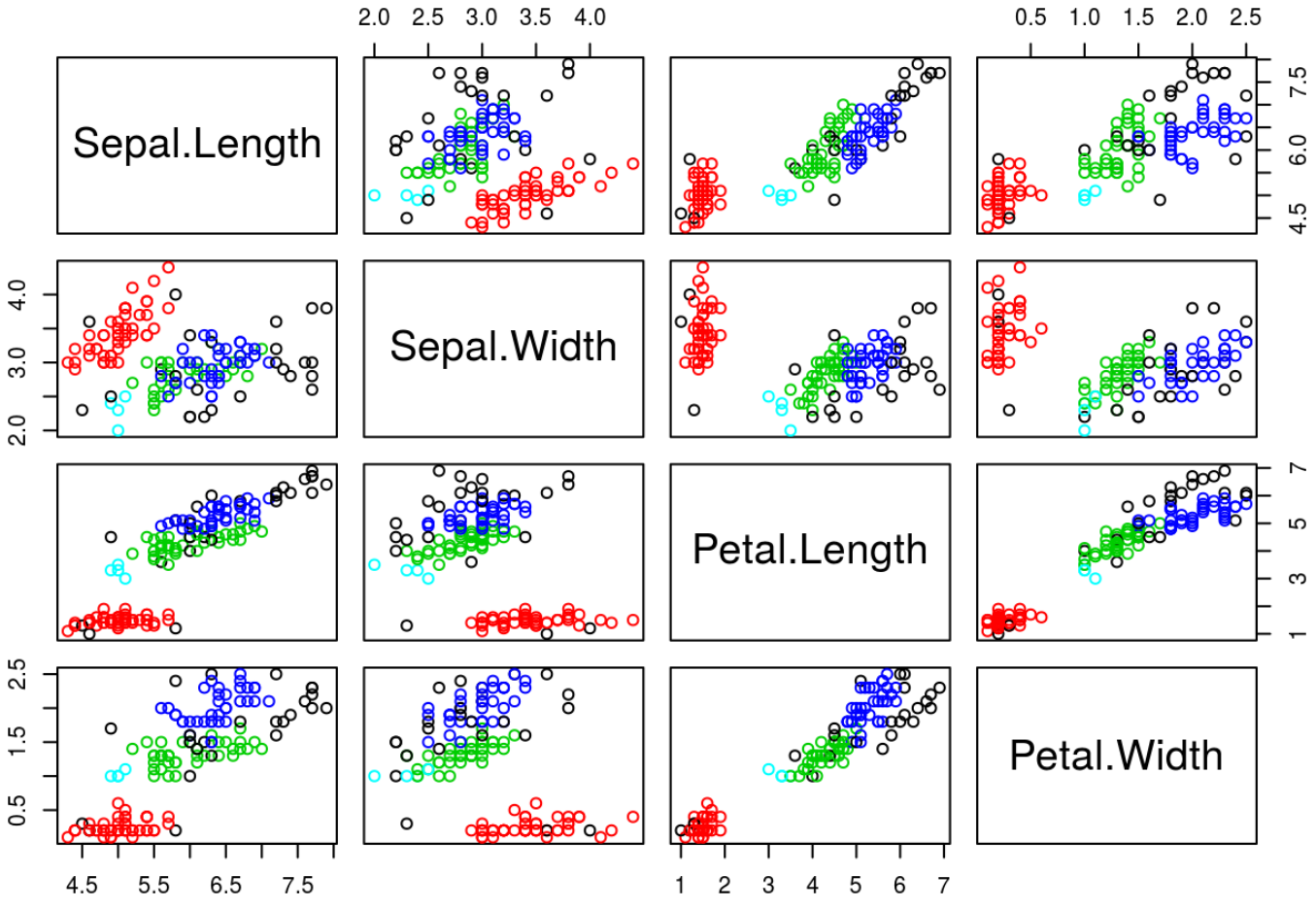
- No exemplo abaixo, comparamos DBSCAN com o *k*-means, através do conjunto de dados simulados.



Exemplo `Iris`

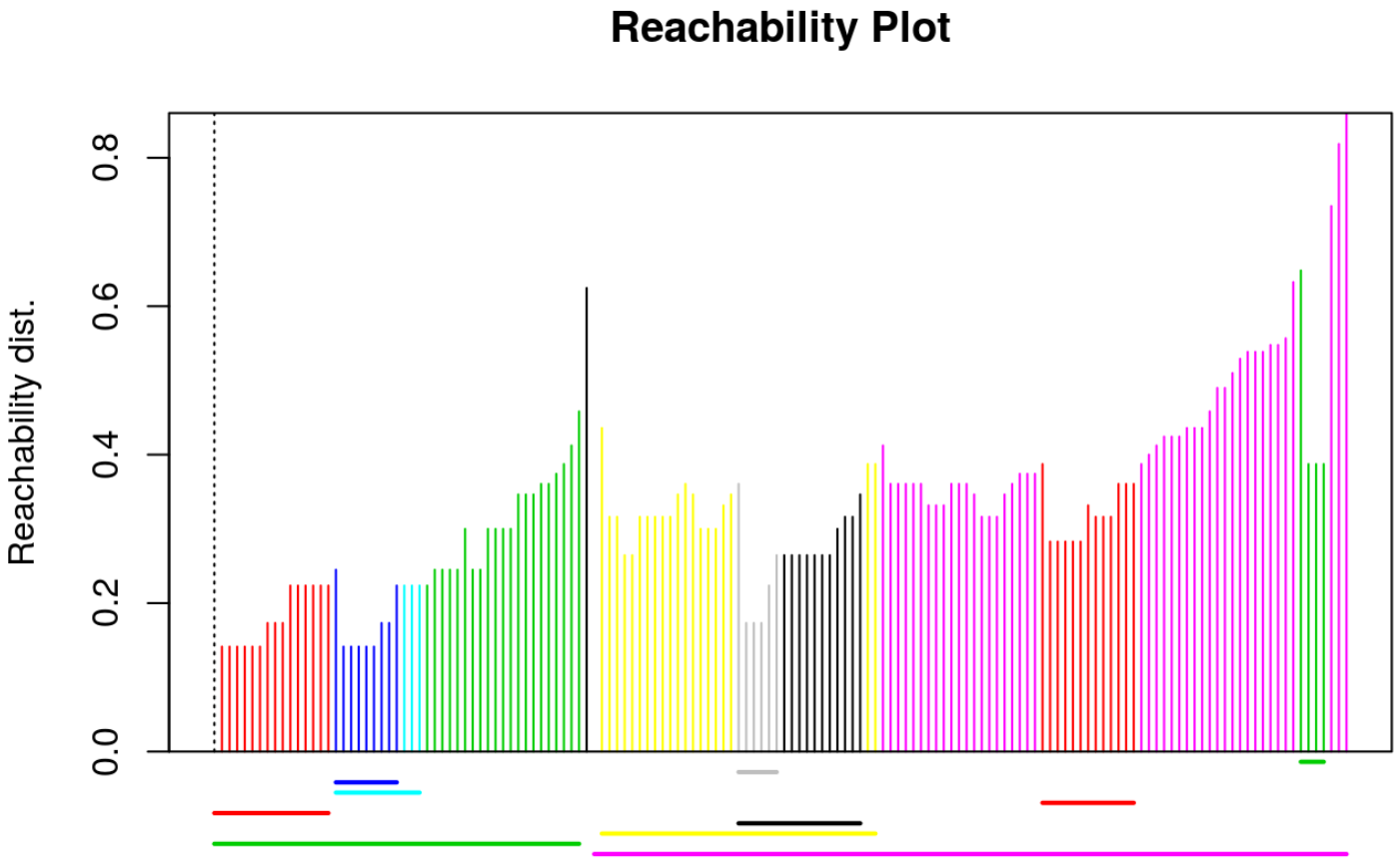


Exemplo Iris

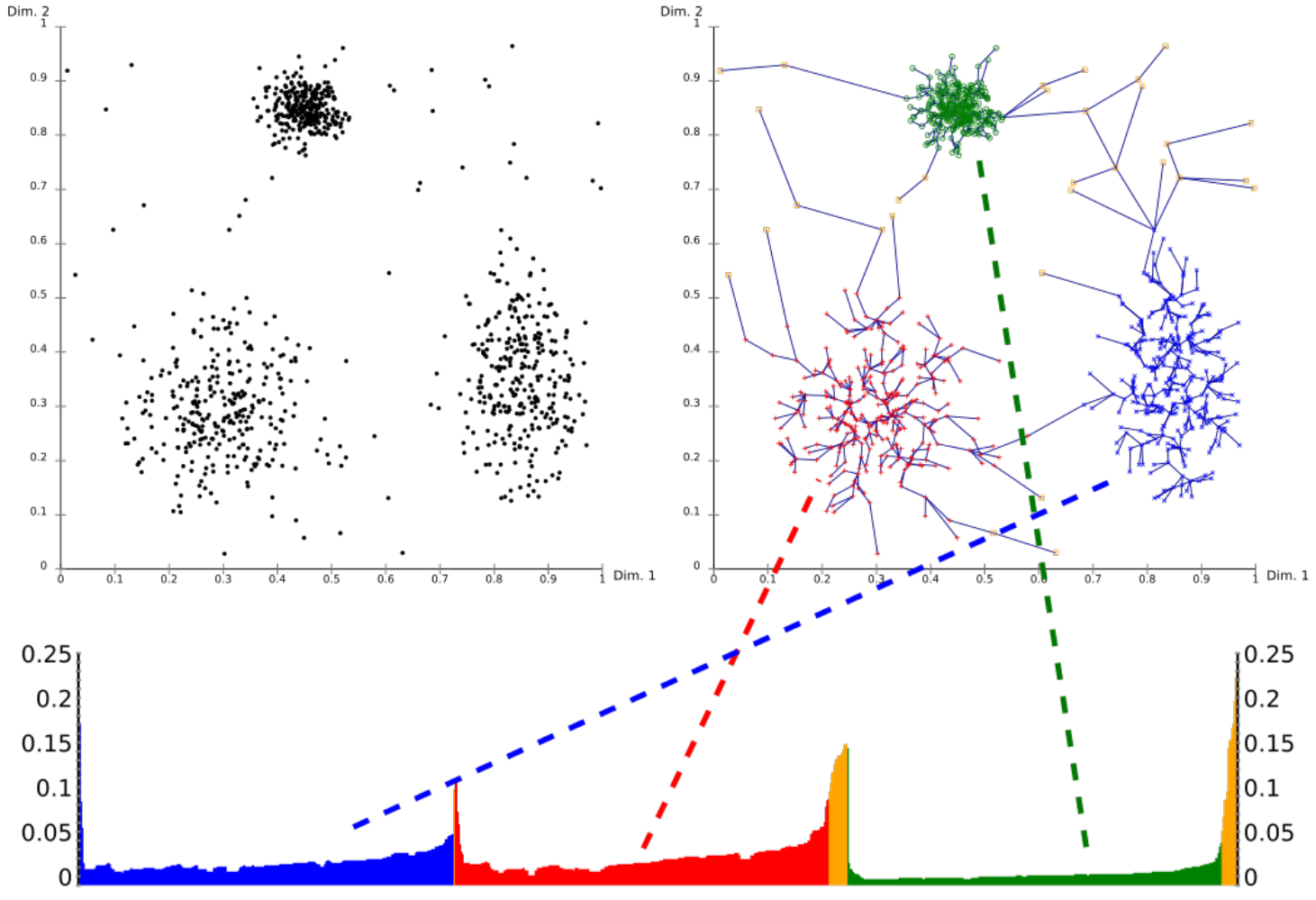


Exemplo Iris

Exemplo Iris



Reachability-plot



Referências

- James, G., Witten, D., Hastie, T. e Tibshirani, An Introduction to Statistical Learning, 2013;
- Hastie, T., Tibshirani, R. e Friedman, J., The Elements of Statistical Learning, 2009;
- Lantz, B., Machine Learning with R, Packt Publishing, 2013;
- Tan, Steinbach, and Kumar, Introduction to Data Mining, Addison-Wesley, 2005;
- Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani