Análise de dados em R

Antes de começar

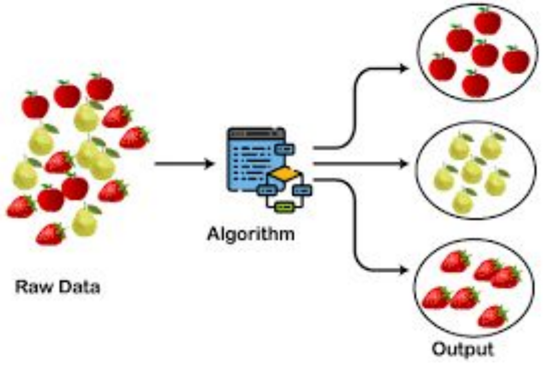
Instalar packages:

```
install.packages(c(“factoextra",”GGally”, “dbscan”))
```

Tasks - Clustering

Grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters).

- Differentiate between different types of tissue in a three-dimensional image for many different purposes
- Create profiles of typical television viewers



Clustering - Step by step

1) Problem definition
2) Data preparation
3) Select clustering algorithm
4) Define clustering algorithm hyper-parameters
5) Sample data
6) Clusters analysis

Clustering - Problem definition

1. Problem definition: Do we need clustering for this problem? What do we want to cluster? Which attributes of our object of study we should consider?

For example, let’s go to starwars dataset. We have a Star wars dataset where which row is a character of the Star Wars films and we have several attributes of the characters.

Clustering - Step by step

2) Data preparation:

- Transform data to the right granularity
- Categorical features to numeric
- calculate features that we want to be use for similarity comparison
- Deal with missing values
- Deal with outliers
- All selected features in same order of magnitude

The goal of tidyr is to help you create tidy data.

Tidy data is data where:

1. **Every column is variable.**
2. **Every row is an observation.**
3. **Every cell is a single value.**

https://tidyr.tidyverse.org/

Clustering - Step by step

3) Select clustering algorithm

There multiple types of clustering algorithms. In this course, we will focus on two:

kmeans
dbscan

Clustering - KMeans

- **Algorithms add the points to k clusters in a manner to decrease the distance to the center, called centroids (average of all points of the cluster), of the cluster**
- **Centroid Based/ Partition Clustering**

1. Define the number of clusters (parameter K).

2. For every point, calculate the Euclidean distance between the point and each of the centroids.

3. Assign the point to its nearest centroid. The points assigned to the same centroid form a cluster.

4. Once clusters are formed, calculate new centroid for each cluster by taking the cluster mean.

5. Repeat step 2, 3 and 4 until the centroids don’t change (converge)

Clustering - DBSCAN

- With algorithm, we know that all observation in a cluster as at least other obs. in the cluster within a define distance (eps).
- Density-based algorithm

1. Define the value of eps and minPts.

2. For each point:

- Calculate its distance from all other points. If the distance is less than or equal to eps then mark that point as a neighbor of x.
- If the point gets a neighboring count greater than or equal to minPts, then mark it as a core point (with number of neighbours higher than minPts) or visited.

3. For each core point, if it not already assigned to a cluster than create a new cluster. Recursively find all its neighboring points and assign them the same cluster as the core point.

4. Continue these steps until all the unvisited points are covered.

Clustering - KMeans vs DBSCAN

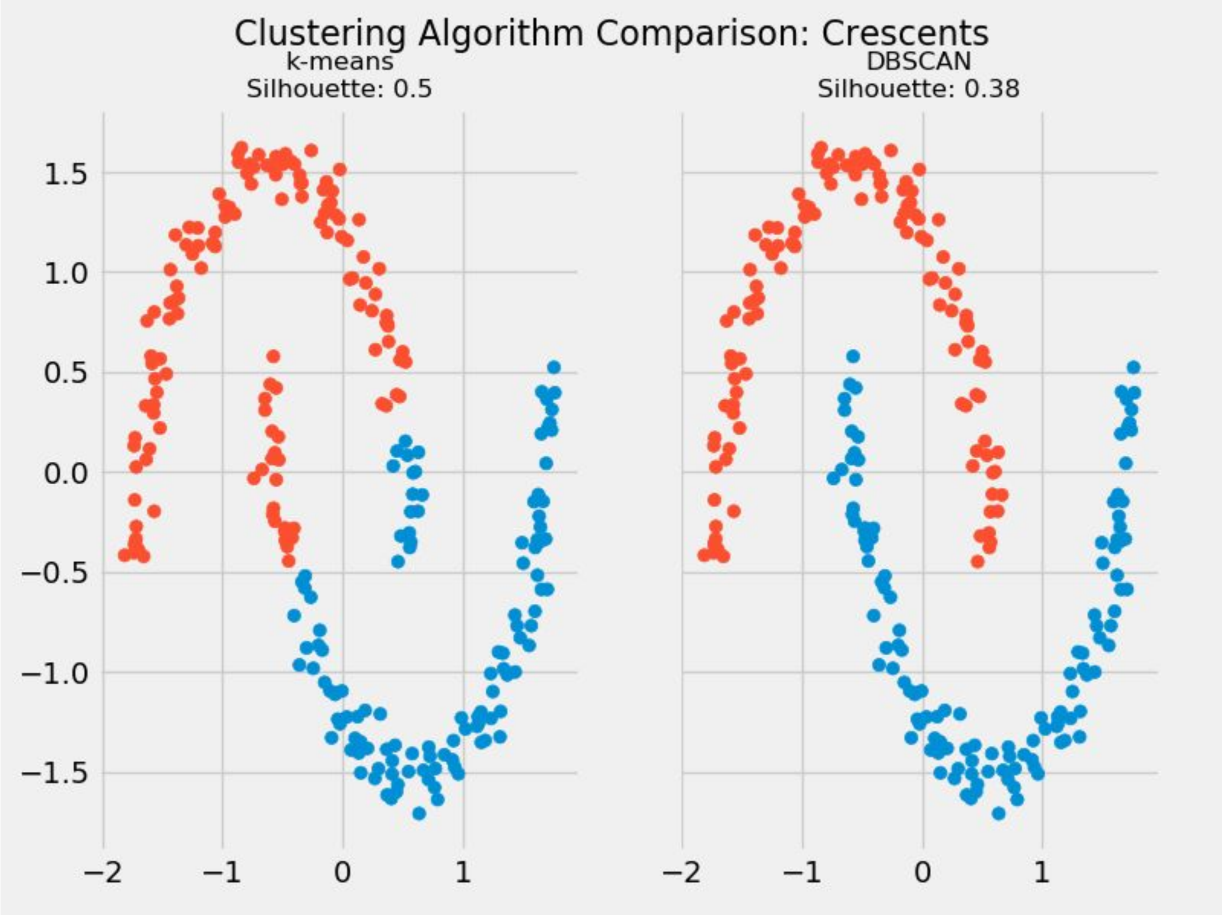
KMEANS

- **Handle large datasets well**
- **Outliers can results with strange results**
- **consider data always centered, can lead strange results**

DBSCAN

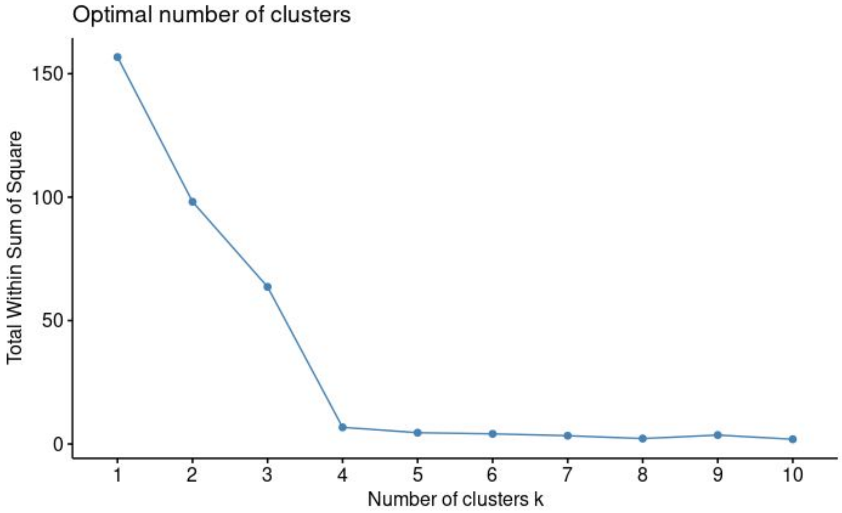
- **Cannot handle with big datasets, it calculates distance between all points**

- **Sensitive to eps and minPts parameters.**

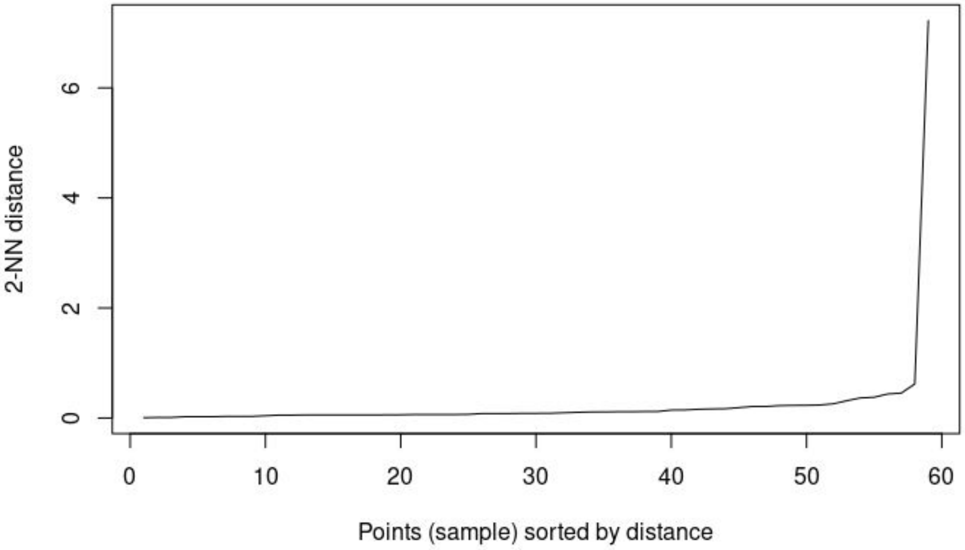

Clustering Algorithm Comparison: Crescents

Clustering - Step by step

4) Define algorithm hyperparameter



Elbow method



kNN method

Clustering - Step by step

6) Clusters analysis

Cluster Yoda (cluster 2)

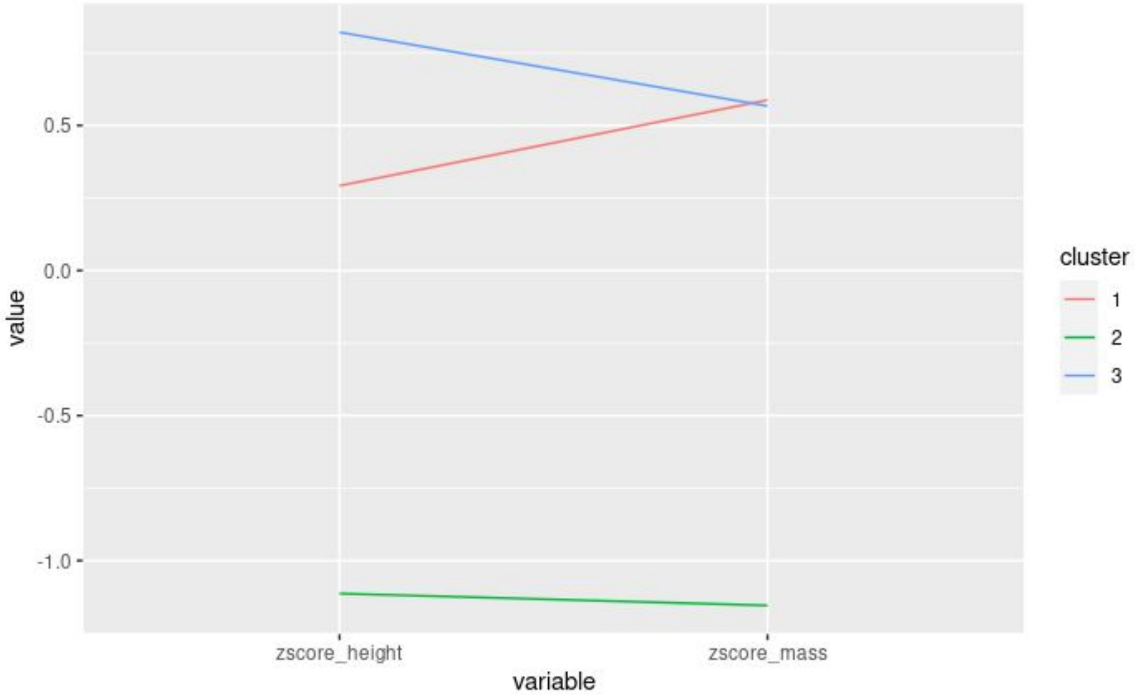
- Small characters

Cluster Tall & Slim (cluster 3)

- Tallest
- and lower mass

Cluster Strong (cluster 1)

- Medium height
- high mass


value
zscore_height
zscore_mass
variable
cluster