Aprendizado Não-Supervisionado

Fabrício Olivetti de França

Universidade Federal do ABC

Tópicos

1. Aprendizado Não-Supervisionado

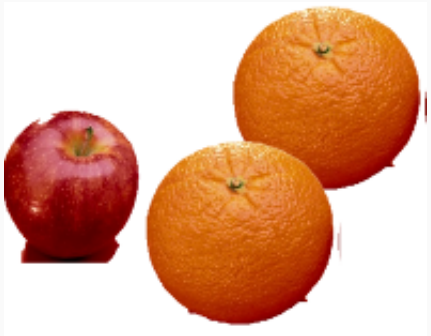
2. Agrupamento de Dados

UFABC

Aprendizado Não-Supervisionado

Aprendizado Não-Supervisionado

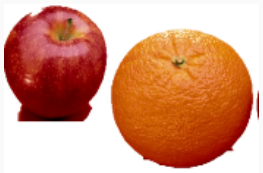
Muitas vezes não temos um rótulo pré-definido que queremos extrair de nossos dados.



Contagem?

1 e 2

Tipo?



UFABC

Aprendizado Não-Supervisionado

O aprendizado não-supervisionado especifica as técnicas para extrair conhecimento de um conjunto de dados sem que tenhamos:

- Informação do que queremos encontrar.
- Qualquer retorno indicando corretude do que encontramos.

Aprendizado Não-Supervisionado

Essas técnicas se baseiam em:

- Formação de grupos de dados similares.
- Modelo de criação de um dado.
- Busca de objetos representativos.
- Redução de Dimensionalidade e Extração de Atributos.

Formação de grupos de dados

Define-se uma medida de similaridade entre nossos dados.

Forme grupos baseados na similaridade entre objetos do mesmo grupo.

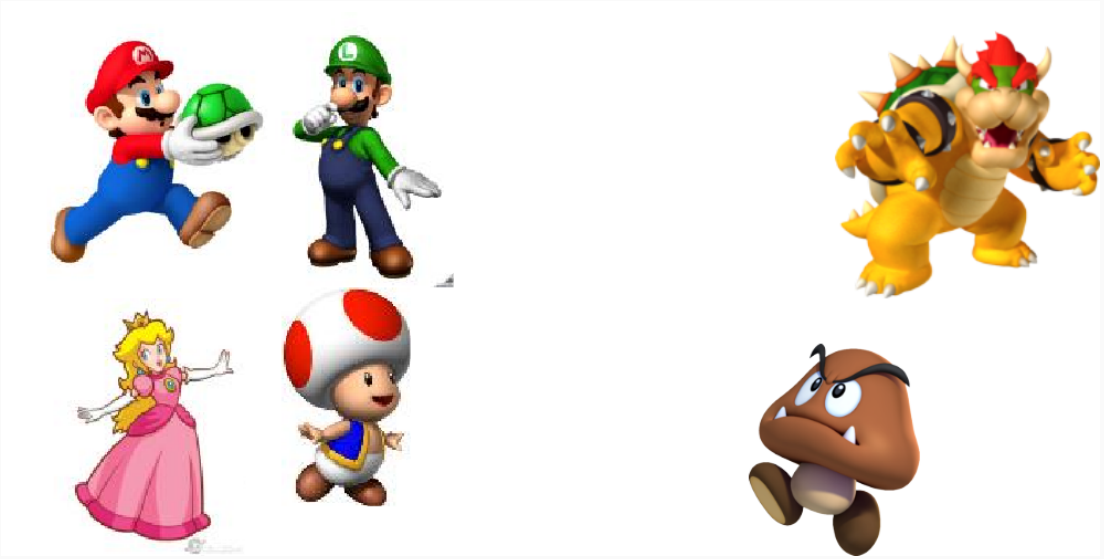
Formação de grupos de dados

Nas imagens abaixo:

UFABC

Formação de grupos de dados

Podemos formar os grupos “bonzinhos” e “malvados”:



UFABC

Formação de grupos de dados

Podemos formar os grupos “masculino”, “feminino” e “indefinido”:

UFABC

Formação de grupos de dados

Podemos formar os grupos “heróis”, “vítimas” e “vilões”:

UFABC

Modelo de Criação

Procura encontrar um modelo que explica a geração dos dados:

$$P(\theta \mid x) = ?$$

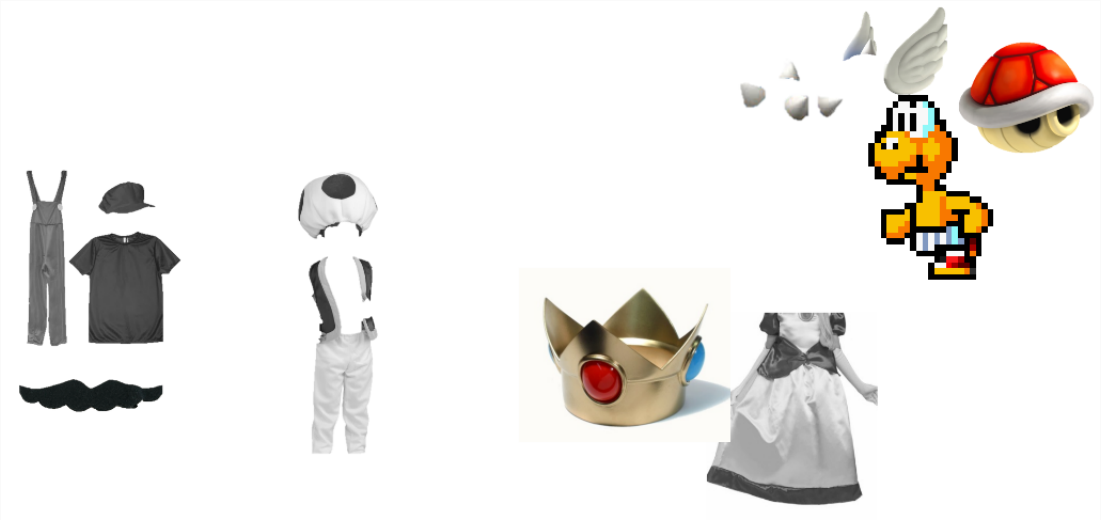
UFABC

Modelo de Criação

Vamos tomar como exemplo agora as seguintes imagens:

Modelo de Criação

Com elas podemos encontrar os seguintes modelos de criação:

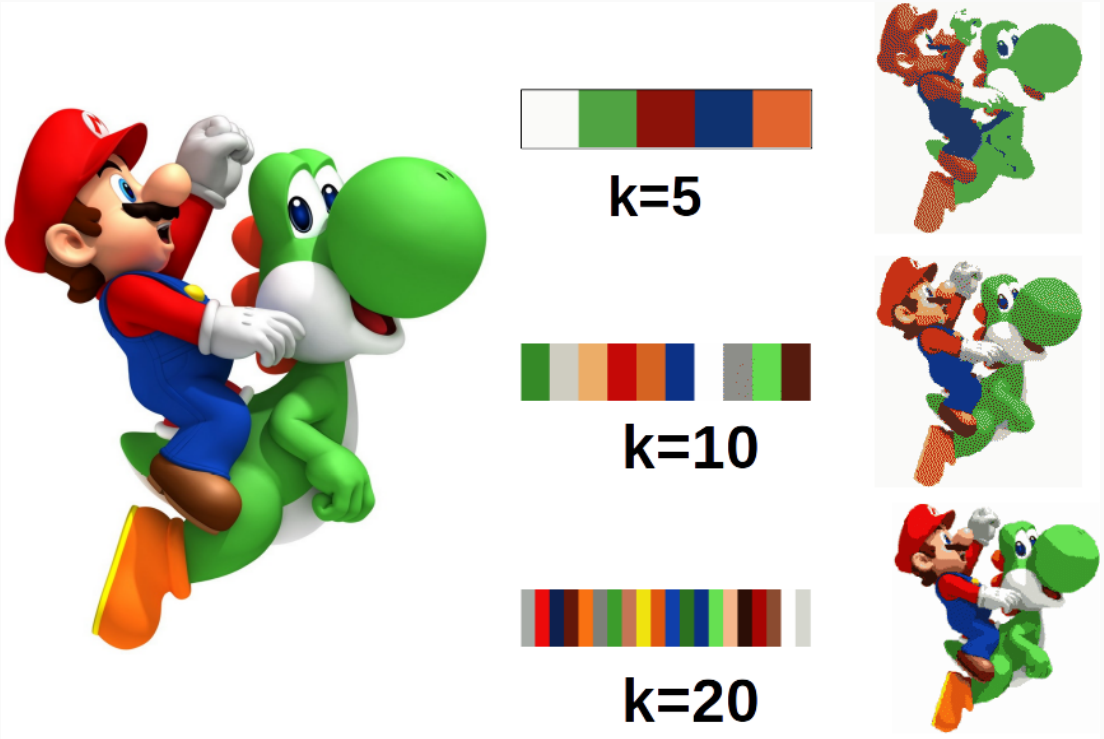


Dados Representativos

Vamos considerar que uma imagem é um conjunto de pixels, e as cores desses pixels são nossos dados.

Se encontrarmos as k cores mais representativa dessa imagem, podemos redesenhá-la utilizando essas cores, economizando bits!

Dados Representativos



UFABC

Redução de Dimensionalidade e Extração de Atributos

Já vimos que em alguns casos (textos, imagens, vídeos) temos a necessidade de extrair atributos estruturados para que seja possível utilizar algoritmos de Aprendizado de Máquina.

Além disso, em algumas situações também pode ser desejável extrair atributos do conjunto de atributos estruturados.

Redução de Dimensionalidade e Extração de Atributos

A extração de novos atributos é motivado por:

- encontrar um espaço de atributos com separação linear entre as classes;
- reduzir a dimensão do problema reduzindo o custo computacional do modelo de aprendizado;
- descobrir relações entre os atributos que podem ser de interesse prático;
- projetar para um espaço bi ou tri-dimensional com o intuito de visualizar os dados.

Redução de Dimensionalidade e Extração de Atributos

Já a redução de atributos é aplicada com o objetivo de minimizar o impacto da *maldição da dimensionalidade*.

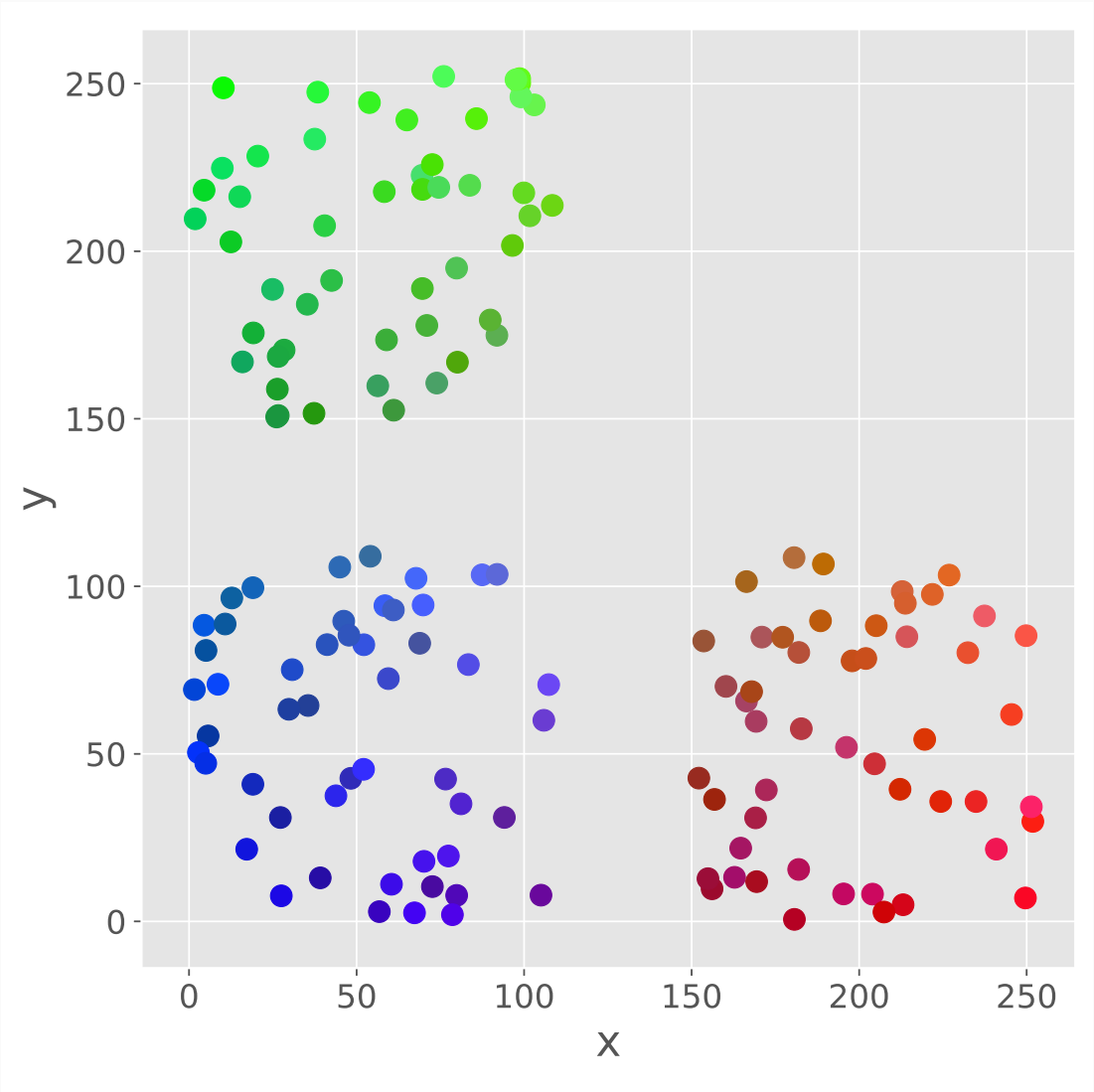
Quando o espaço de atributos tem uma dimensão muito grande, observamos alguns fenômenos:

- A diferença entre as distâncias de objetos similares e dissimilares se torna pequena.
- Muitos atributos não contribuem com o objetivo do aprendizado.
- Alguns atributos inserem ruídos que atrapalham o ajuste dos parâmetros livres dos algoritmos.

Agrupamento de Dados

k-Means

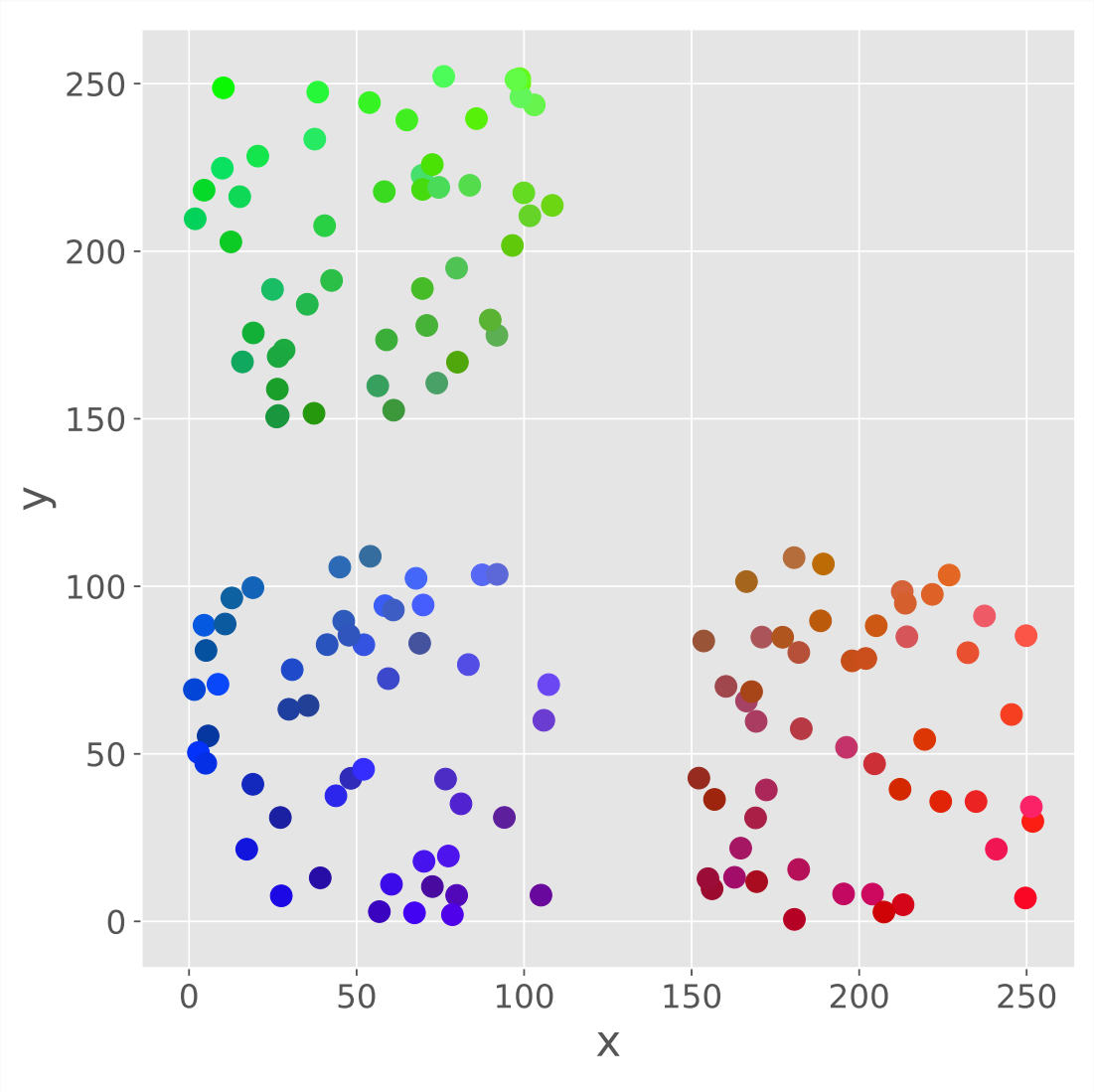
Digamos que temos diversas cores que são tons de vermelho, verde e azul.



UFABC

k-Means

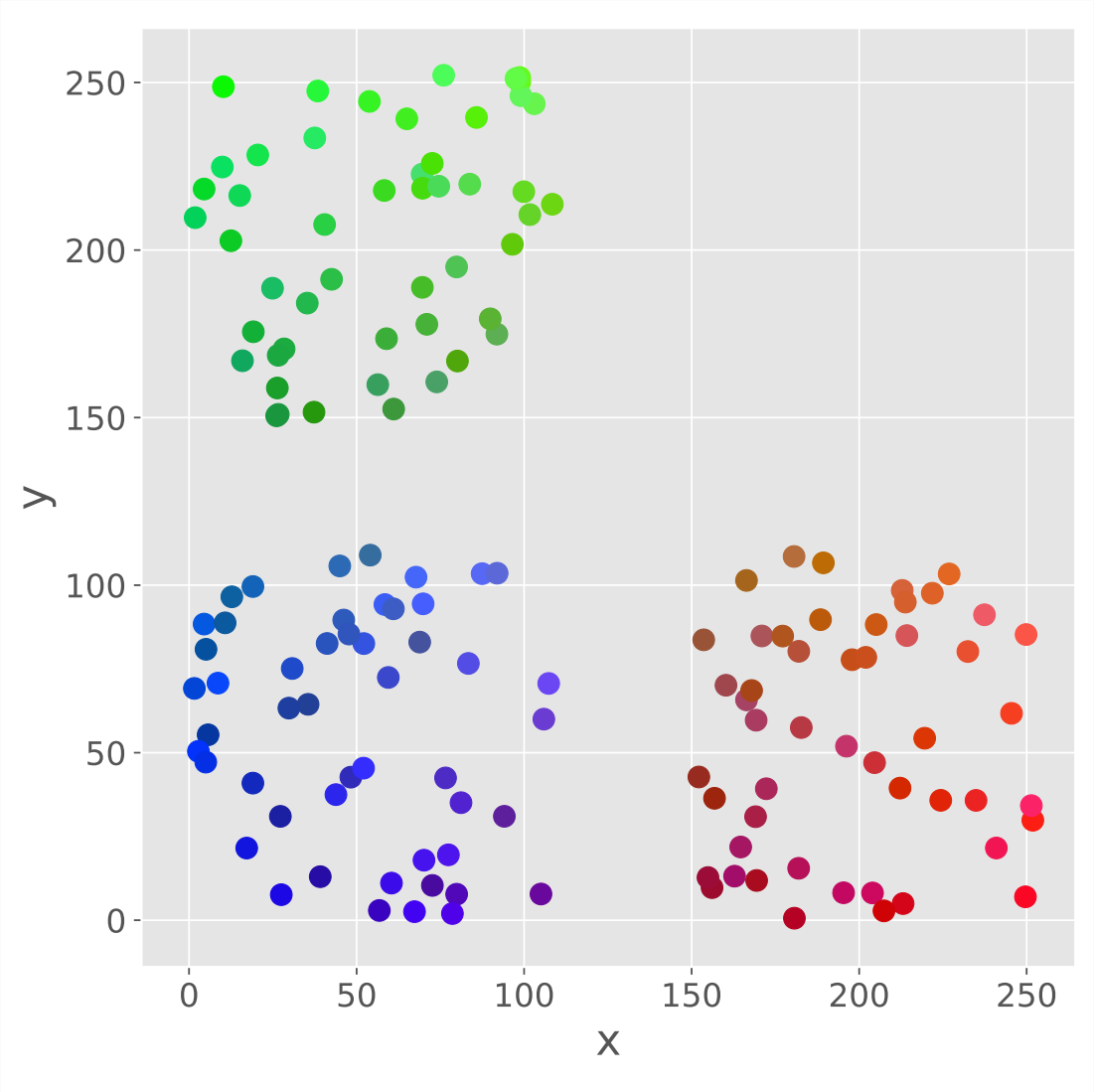
Queremos escolher as 3 cores mais representativas desse conjunto!



UFABC

k-Means

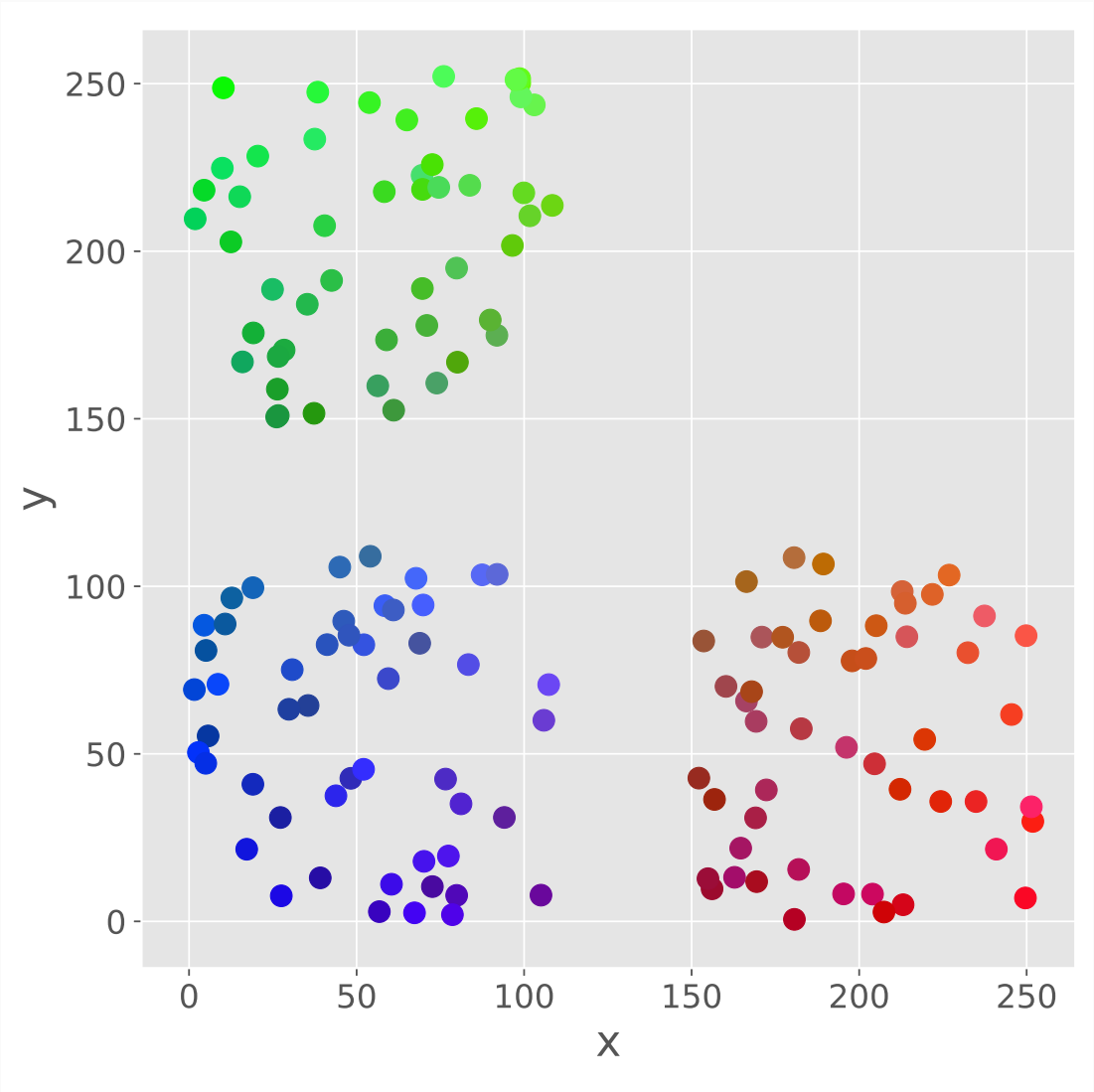
E não necessariamente essas 3 cores devem fazer parte do conjunto de dados, podemos criar cores novas!



UFABC

k-Means

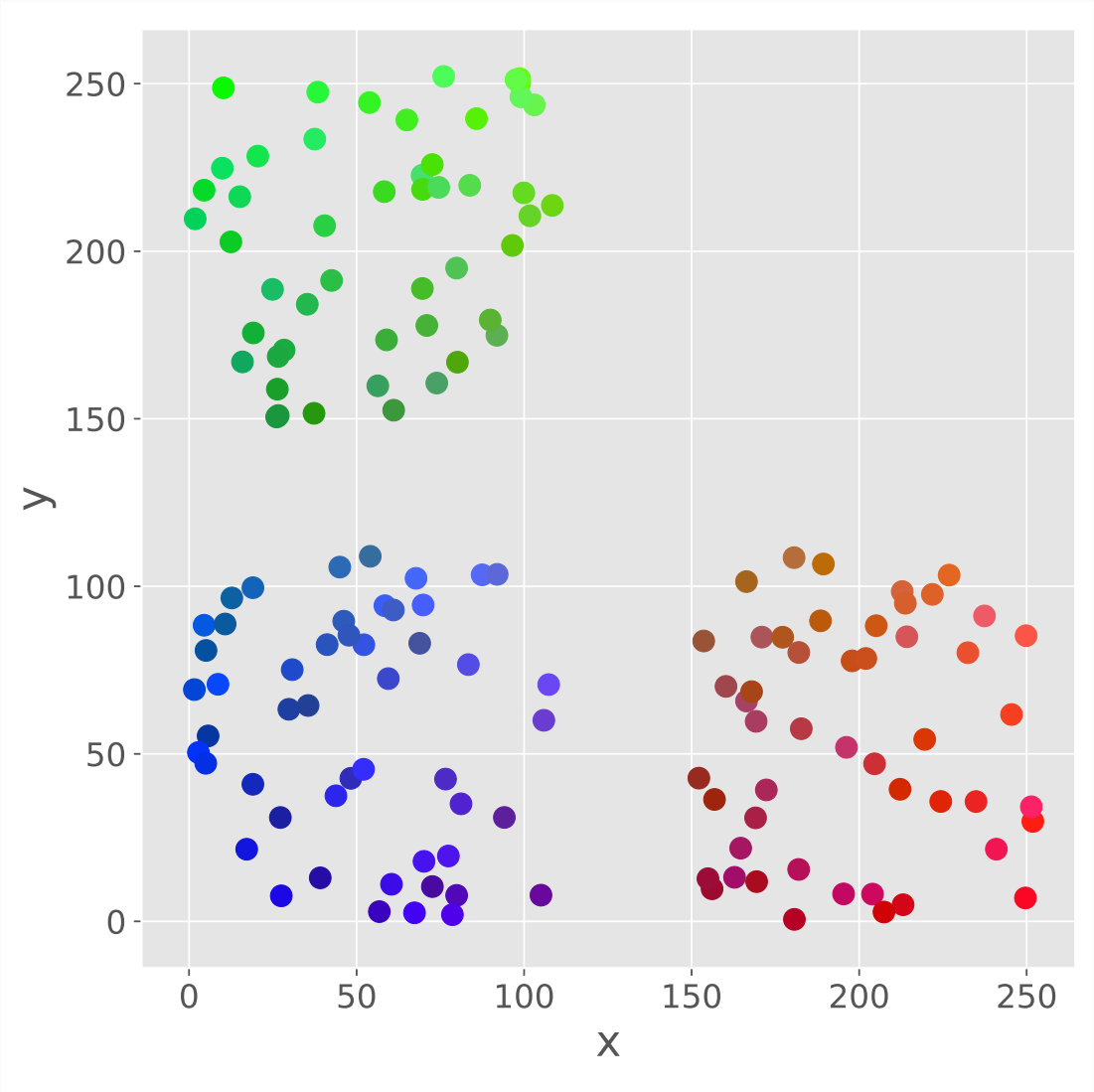
O que caracteriza uma cor representativa de um conjunto?



UFABC

k-Means

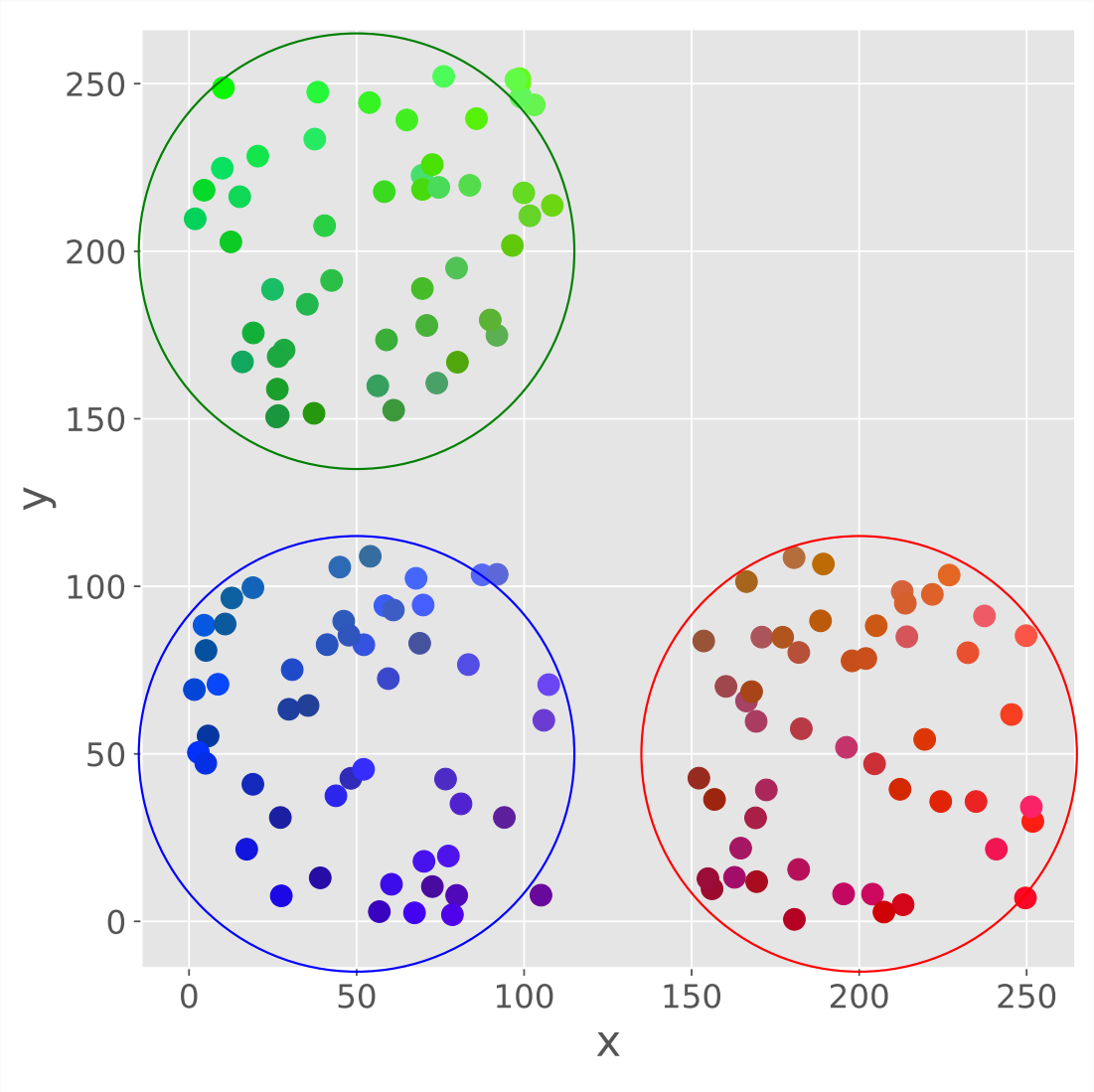
Bom, para saber isso primeiro precisamos conhecer os conjuntos!



UFABC

k-Means

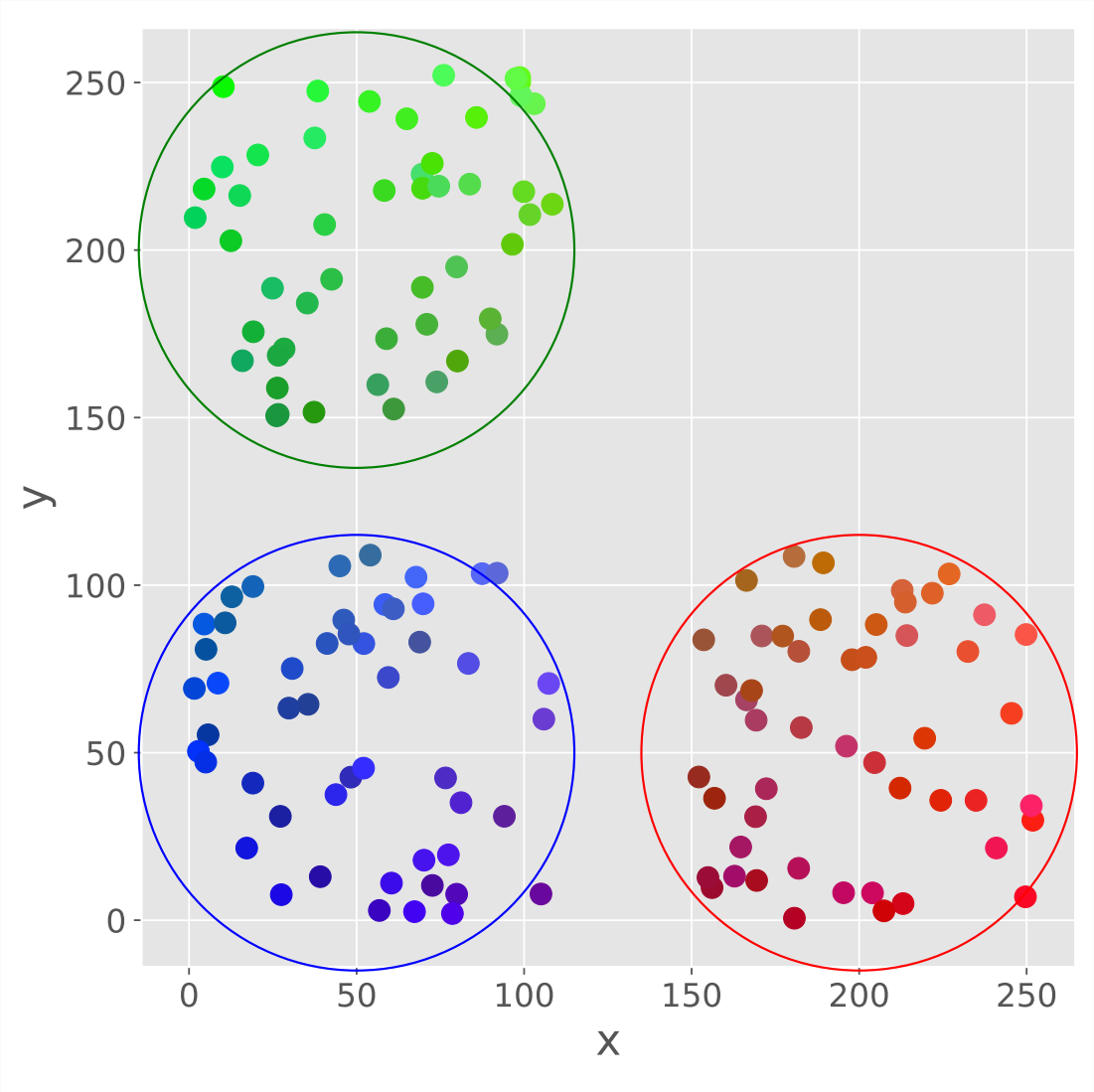
Para nosso exemplo é fácil:



UFABC

k-Means

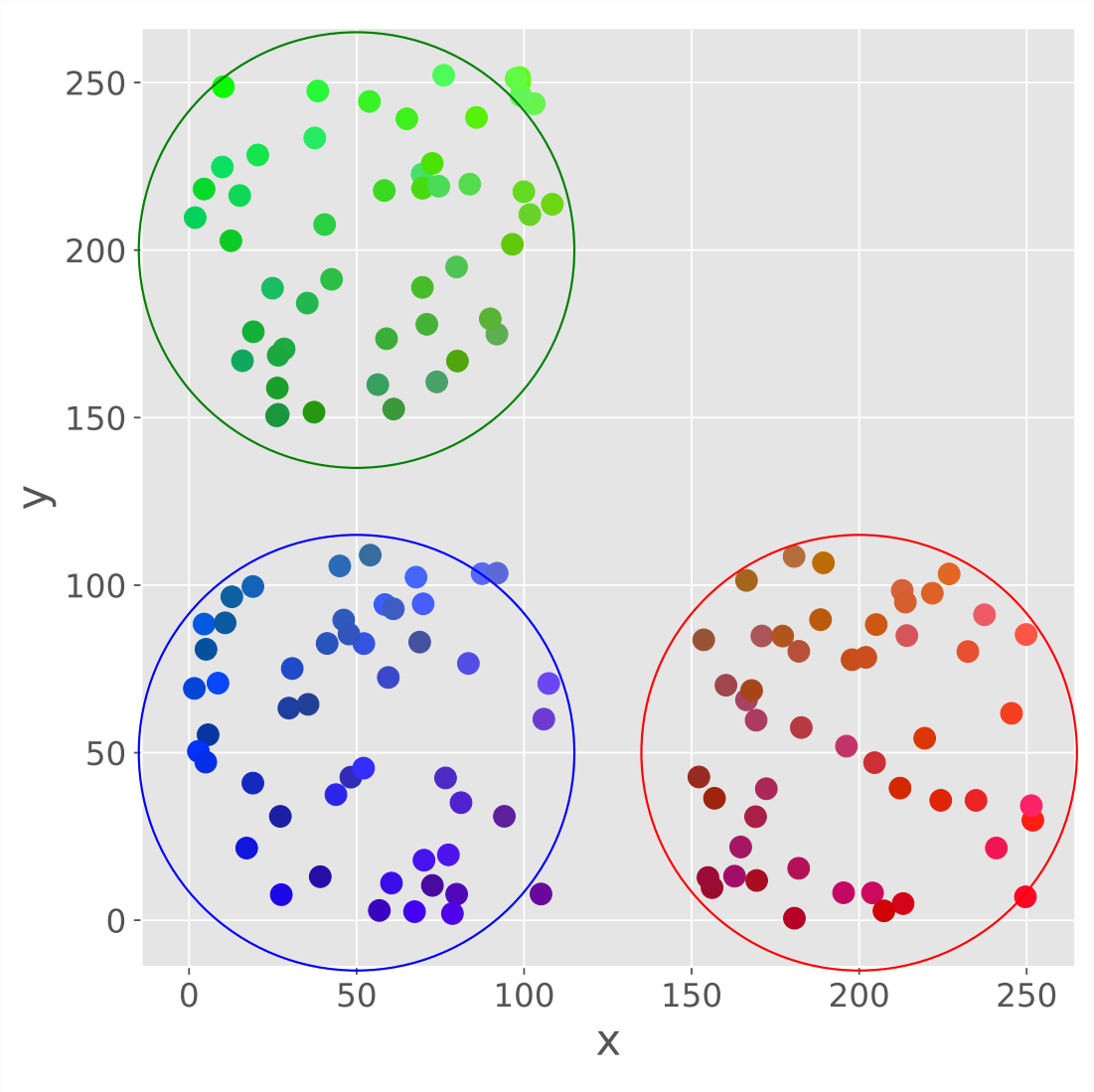
A cor representativa de cada grupo é aquela que mais se aproxima de todas do grupo.



UFABC

k-Means

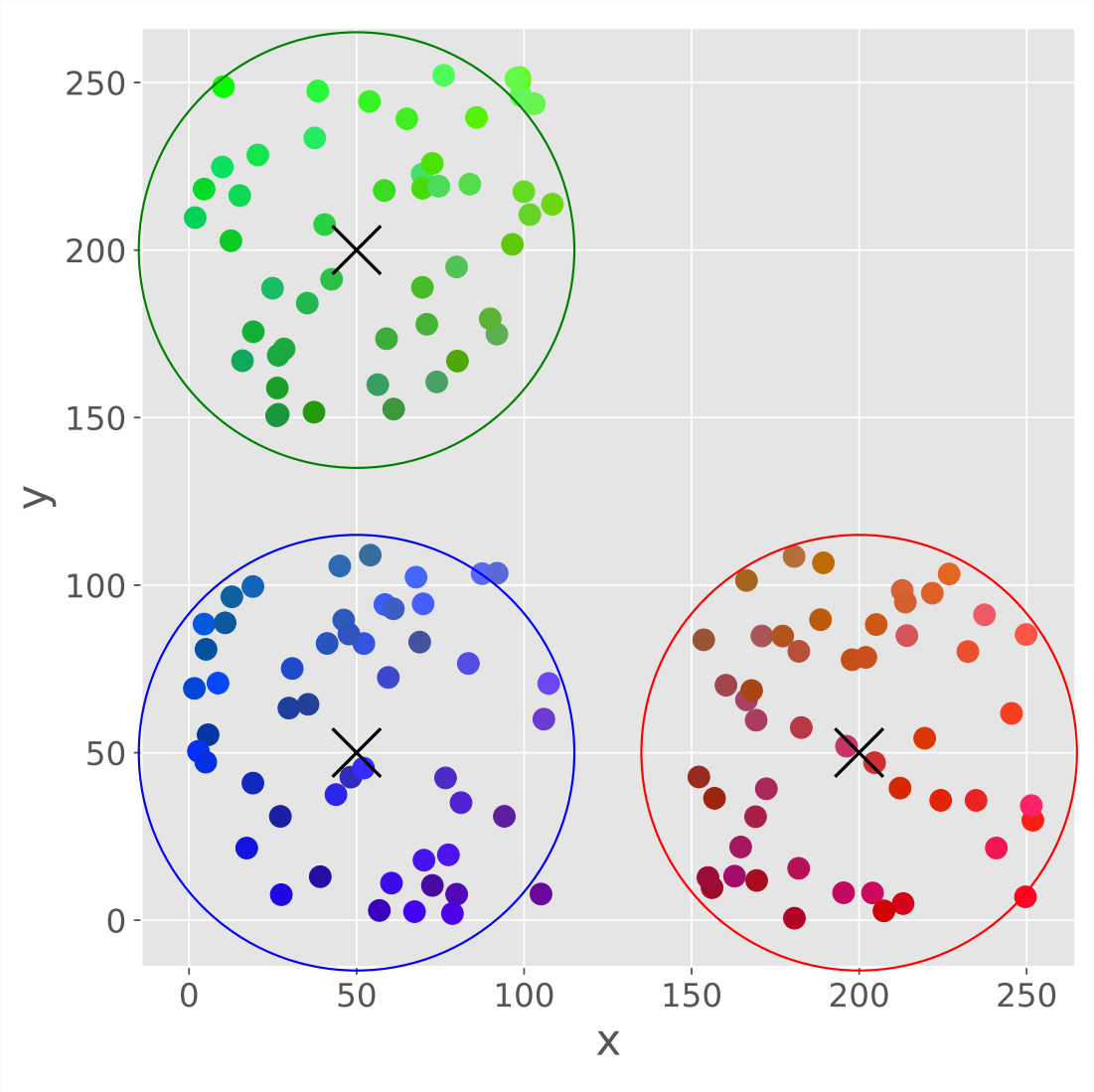
Ou seja, é a que tem maior similaridade média com os elementos do grupo.



UFABC

k-Means

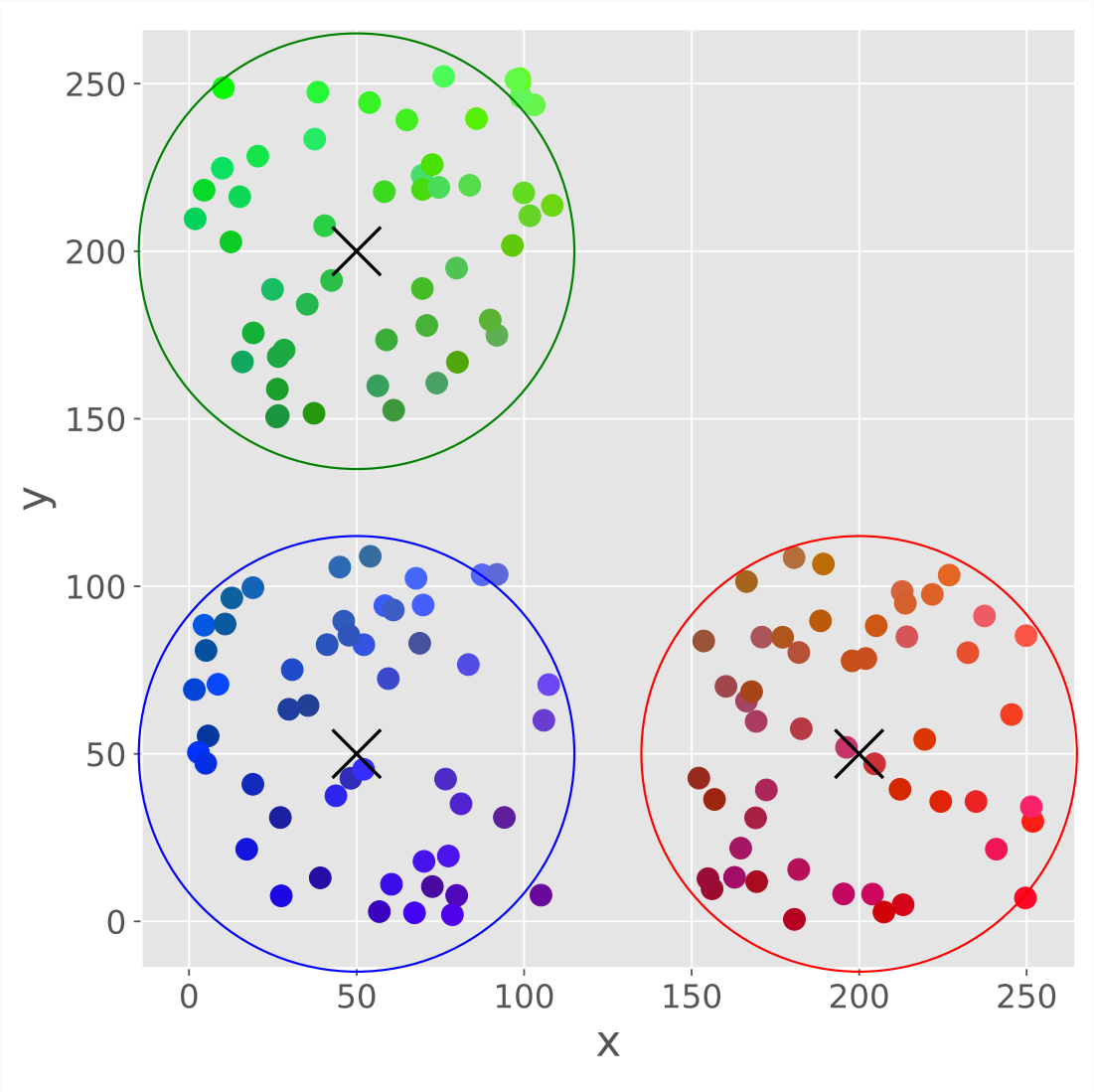
Ou seja, é a que tem maior similaridade média com os elementos do grupo.



UFABC

k-Means

Matematicamente, quem são esses elementos?



UFABC

k-Means

Dado um conjunto de pontos X n-dimensionais, queremos determinar o ponto c, representando o centro, que minimiza:

$$J(X, \mathbf{c}) = \frac{1}{m} \sum_{i=1}^{m} dist(\mathbf{x_i}, \mathbf{c})$$

definimos que a distância é o inverso da similaridade.

k-Means

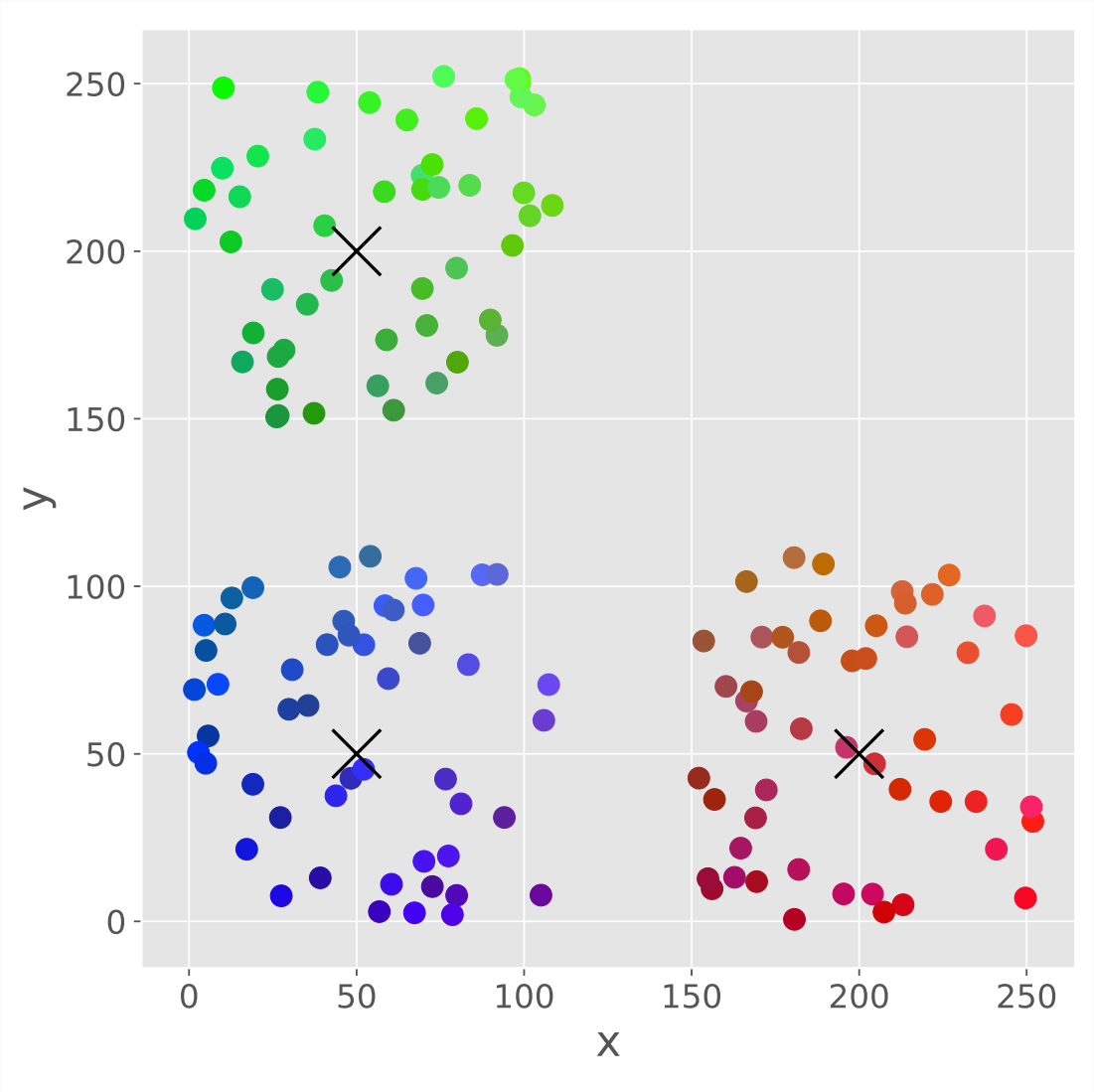
O mínimo pode ser encontrado com:

$$\nabla J(X, \mathbf{c}) = \frac{1}{m} \sum_{i=1}^{m} \nabla dist(\mathbf{x_i}, \mathbf{c}) = 0$$

UFABC

k-Means

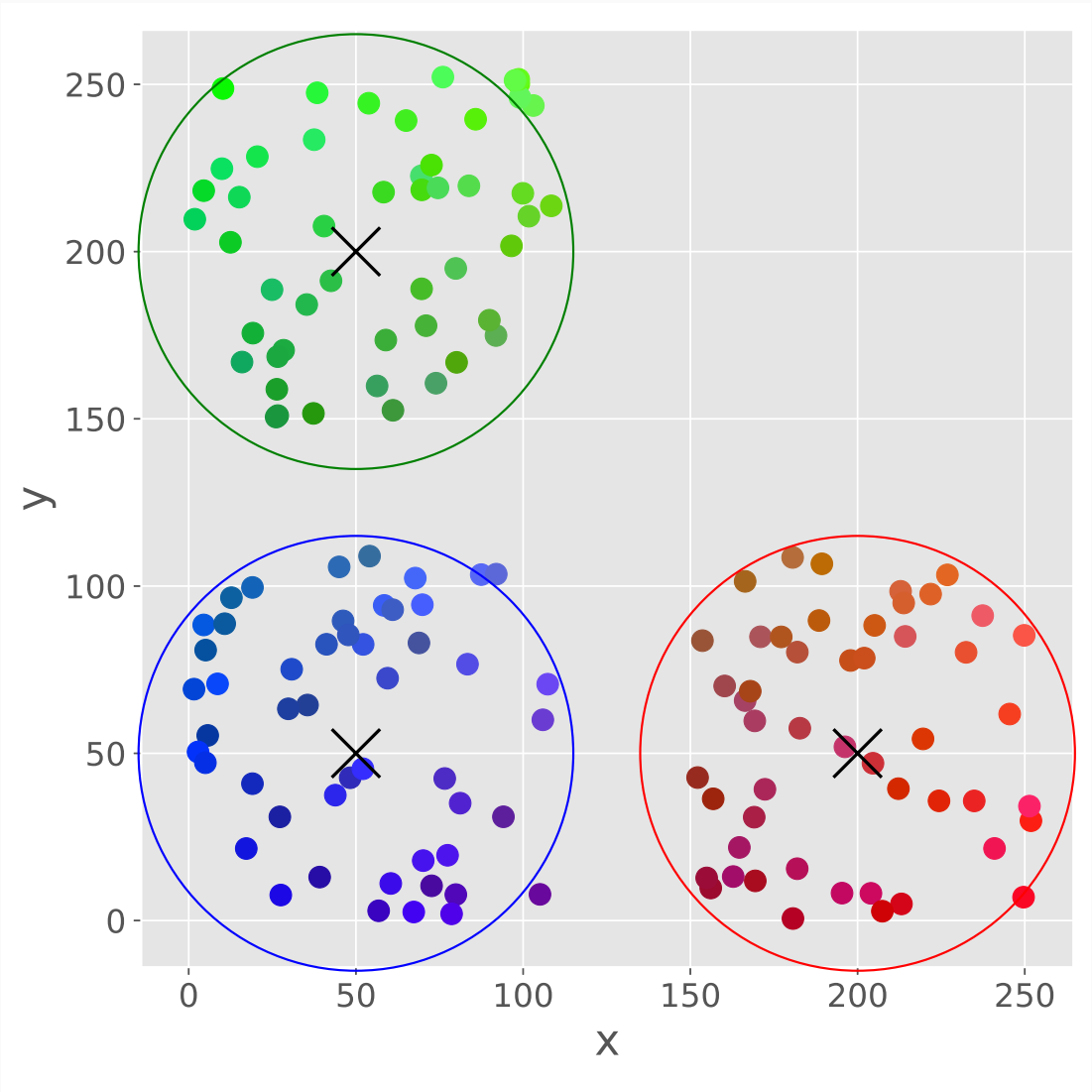
Temos os centros...como definir os grupos?



UFABC

k-Means

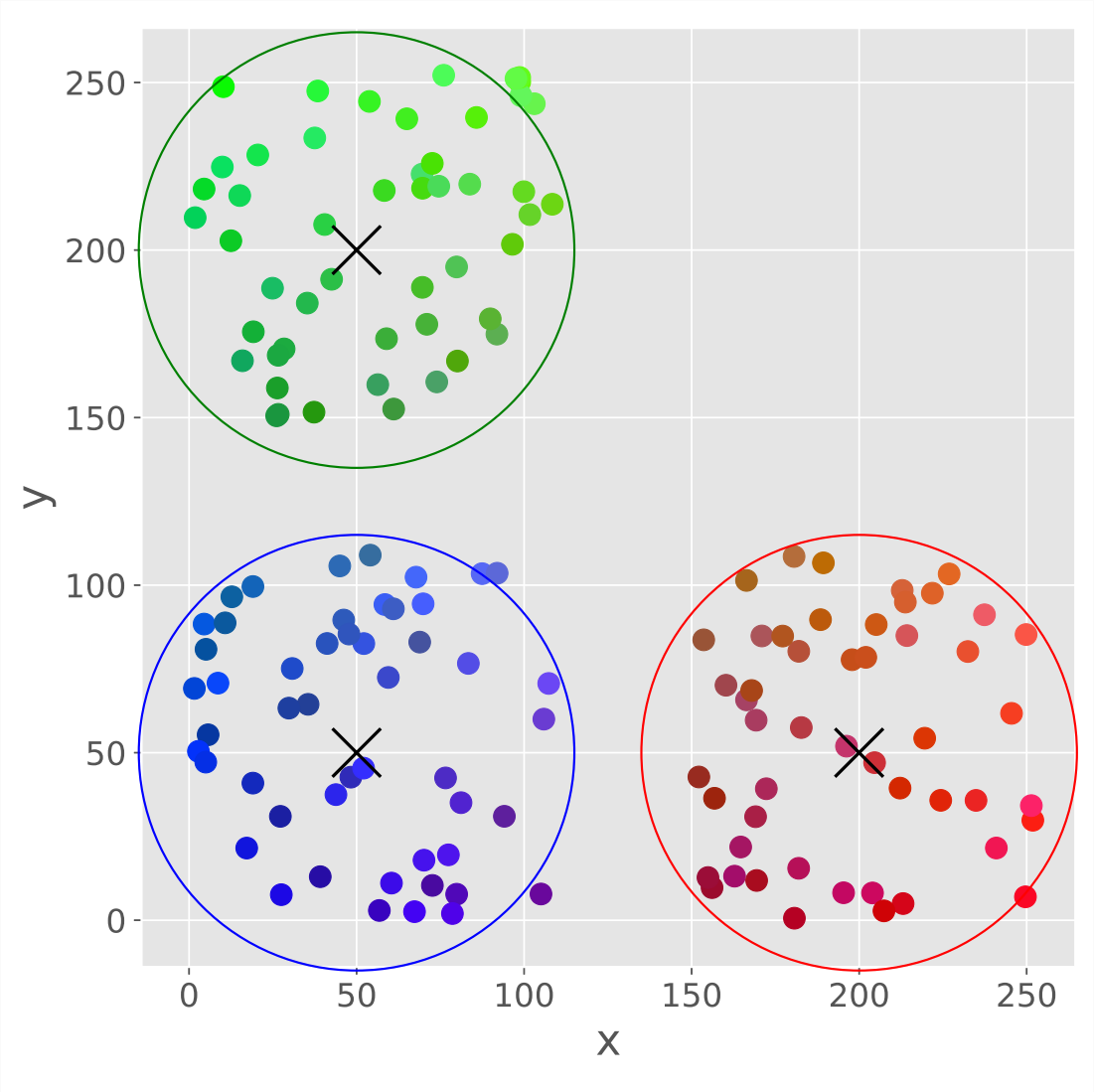
Para cada amostra x, verificamos o centro mais próximo...essa amostra fará parte desse grupo.



UFABC

k-Means

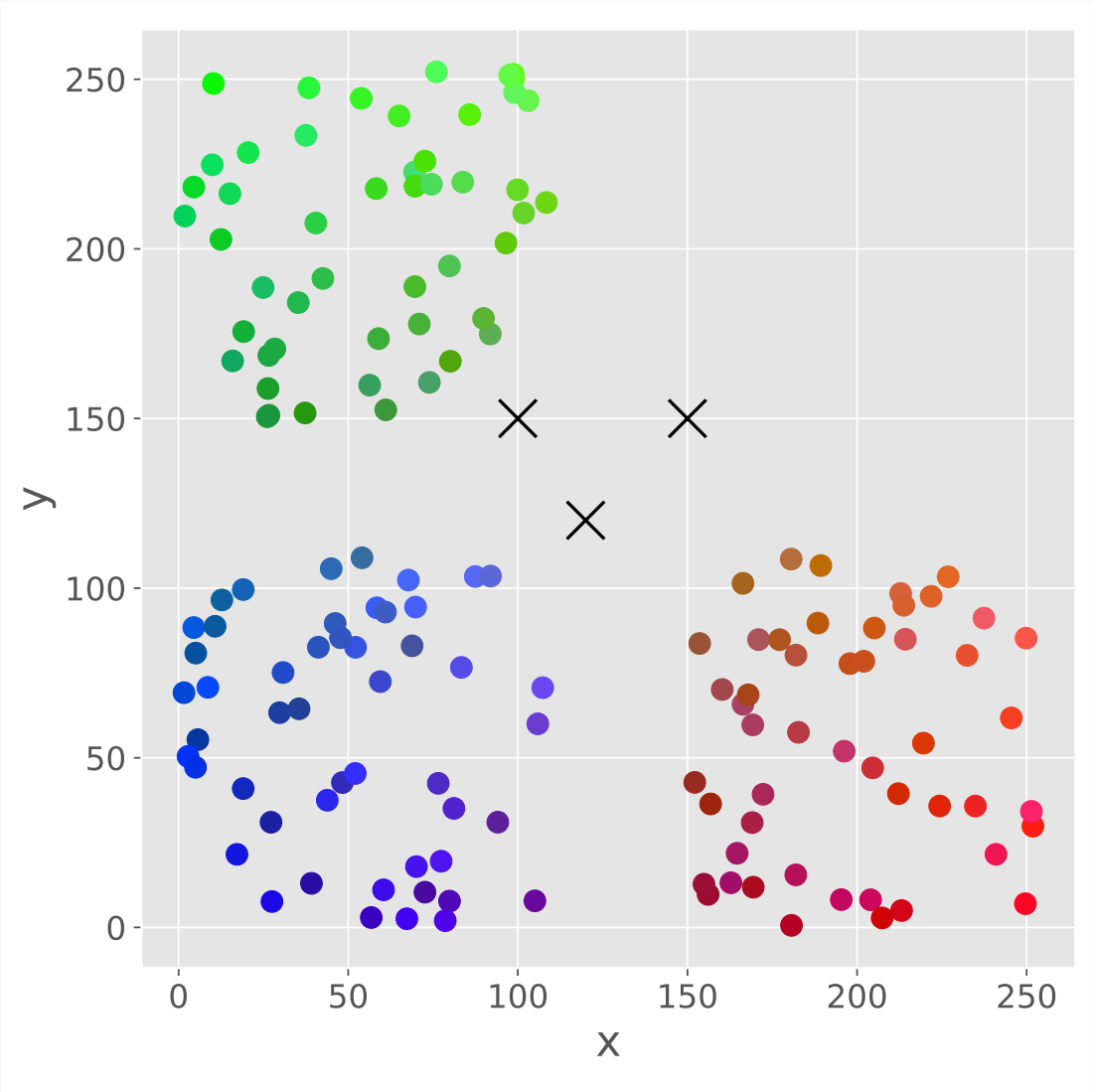
Sabemos como calcular o centro, dado os grupos...e como definir os grupos dado os centros.



UFABC

k-Means

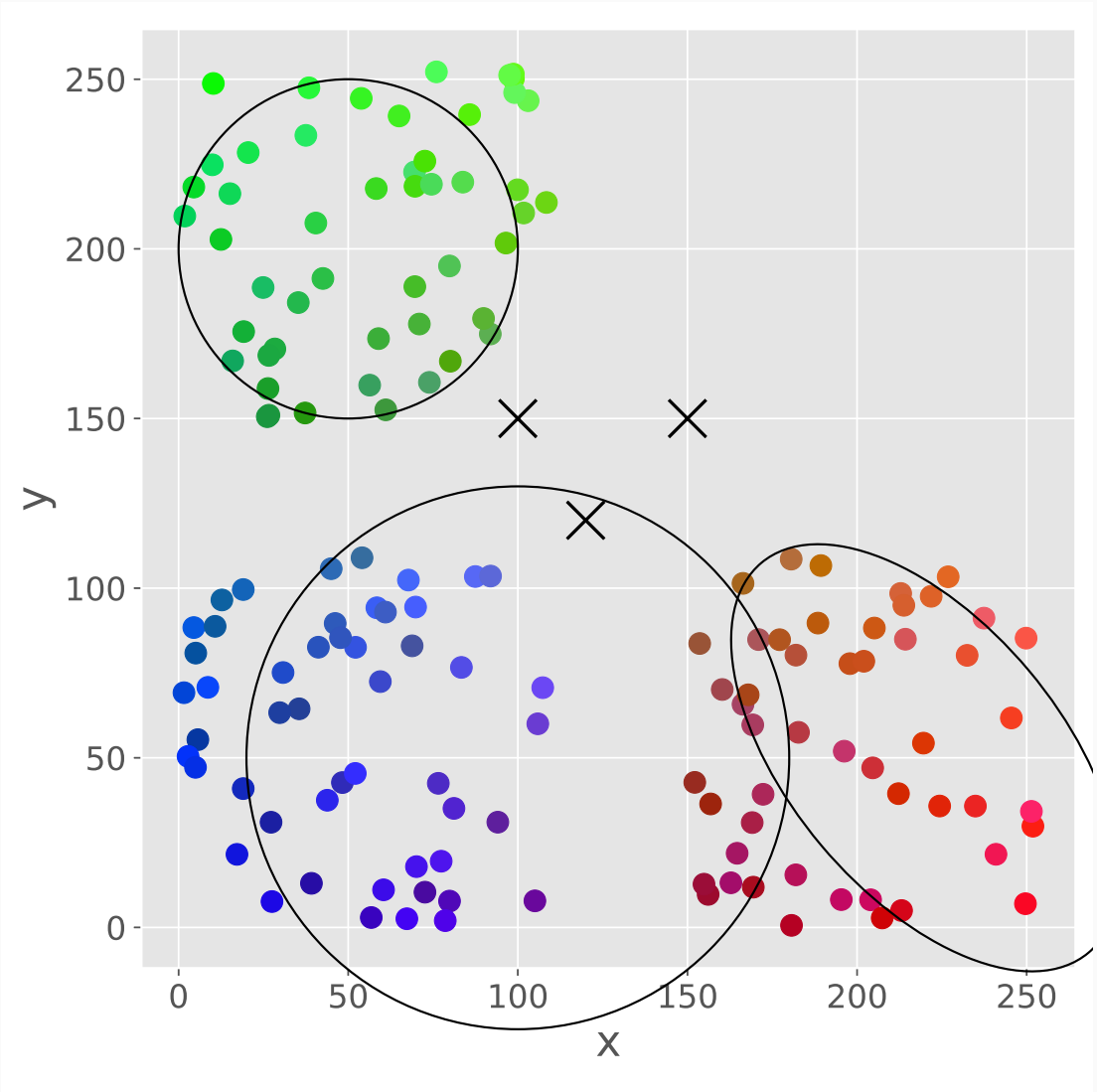
Começamos chutando pontos iniciais para os centros.



UFABC

k-Means

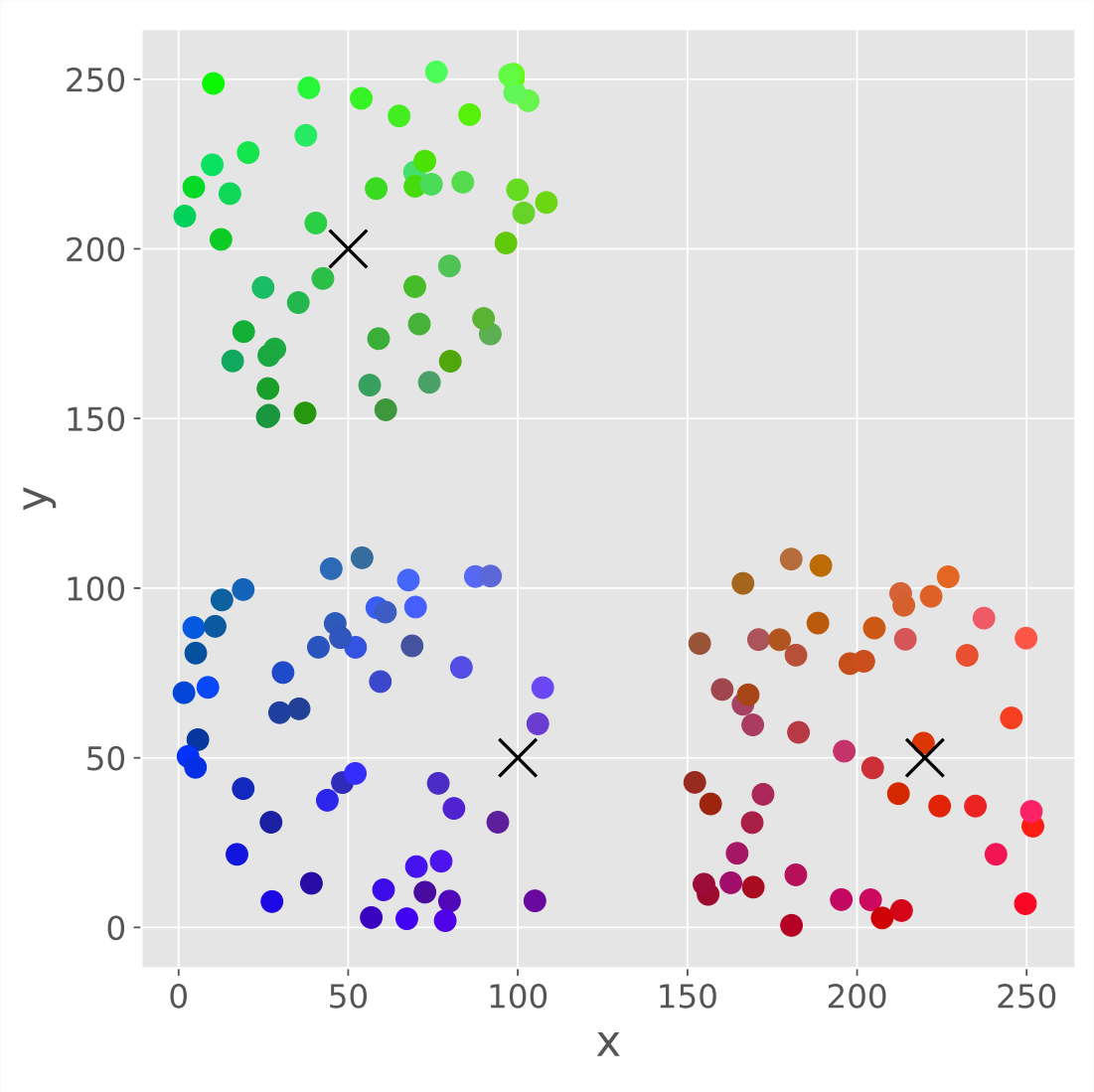
Definimos os grupos de acordo com esses centros:



UFABC

k-Means

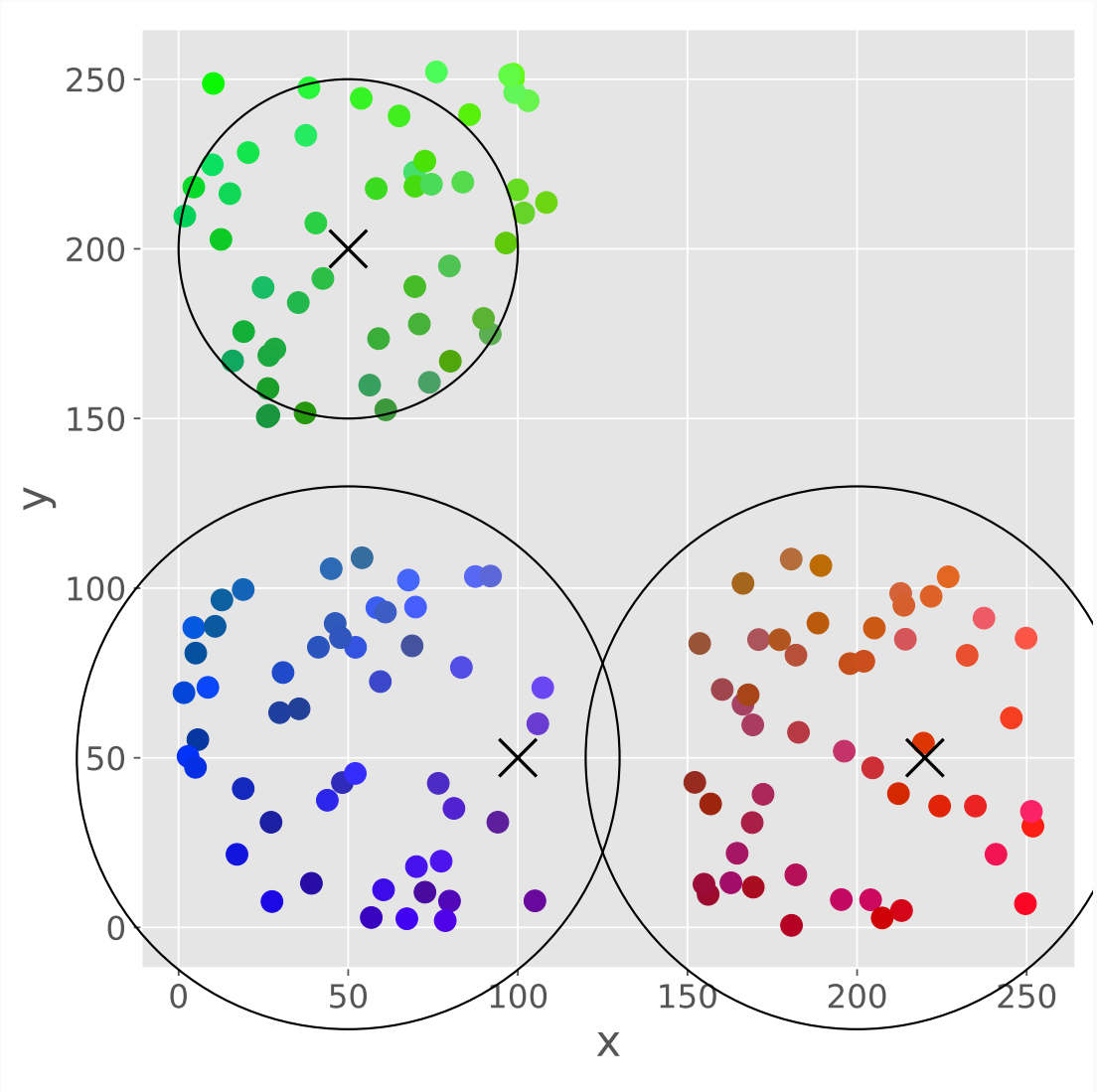
Para cada grupo, calculamos o novo centro:



UFABC

k-Means

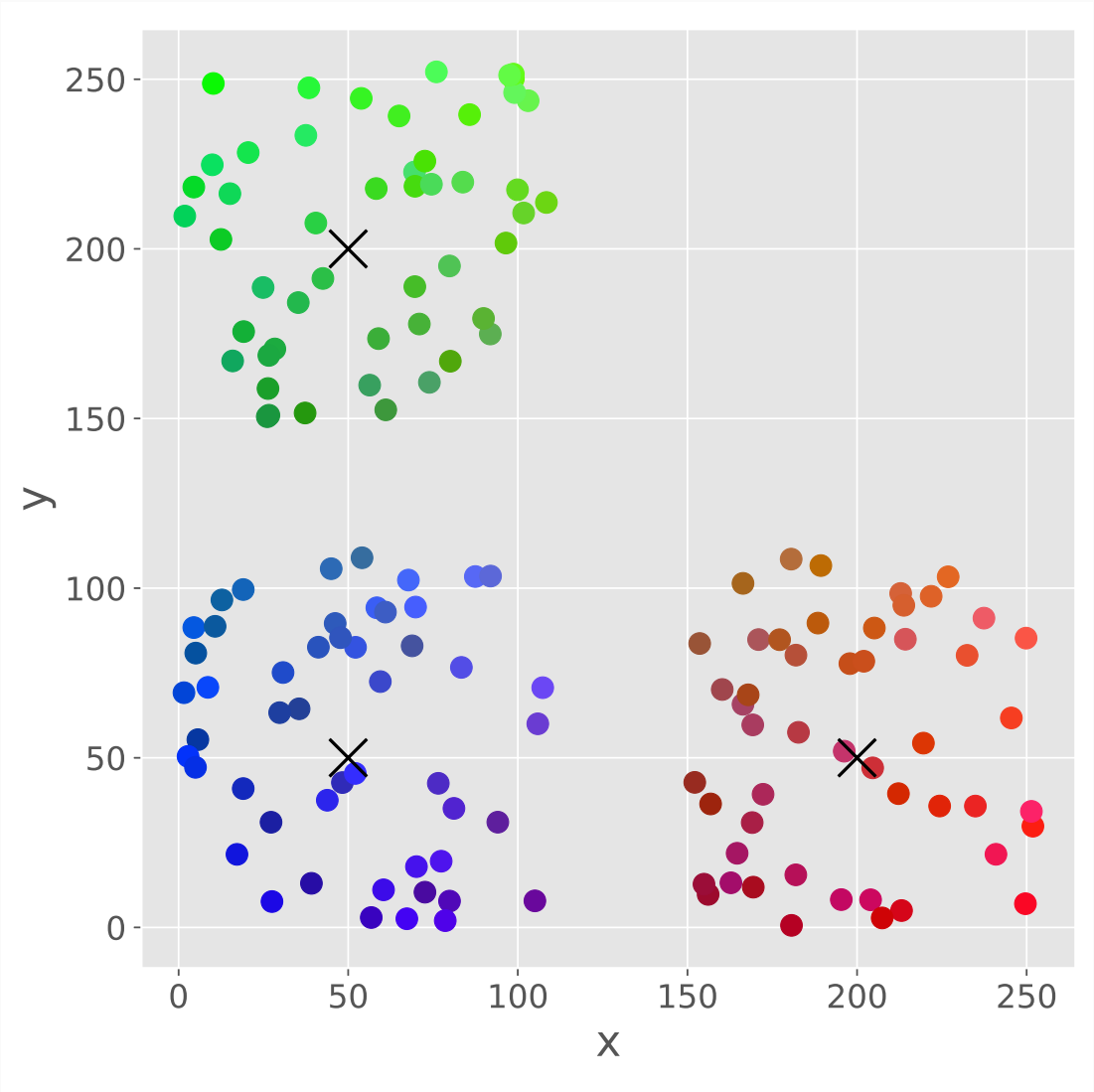
E remontamos os grupos:



UFABC

k-Means

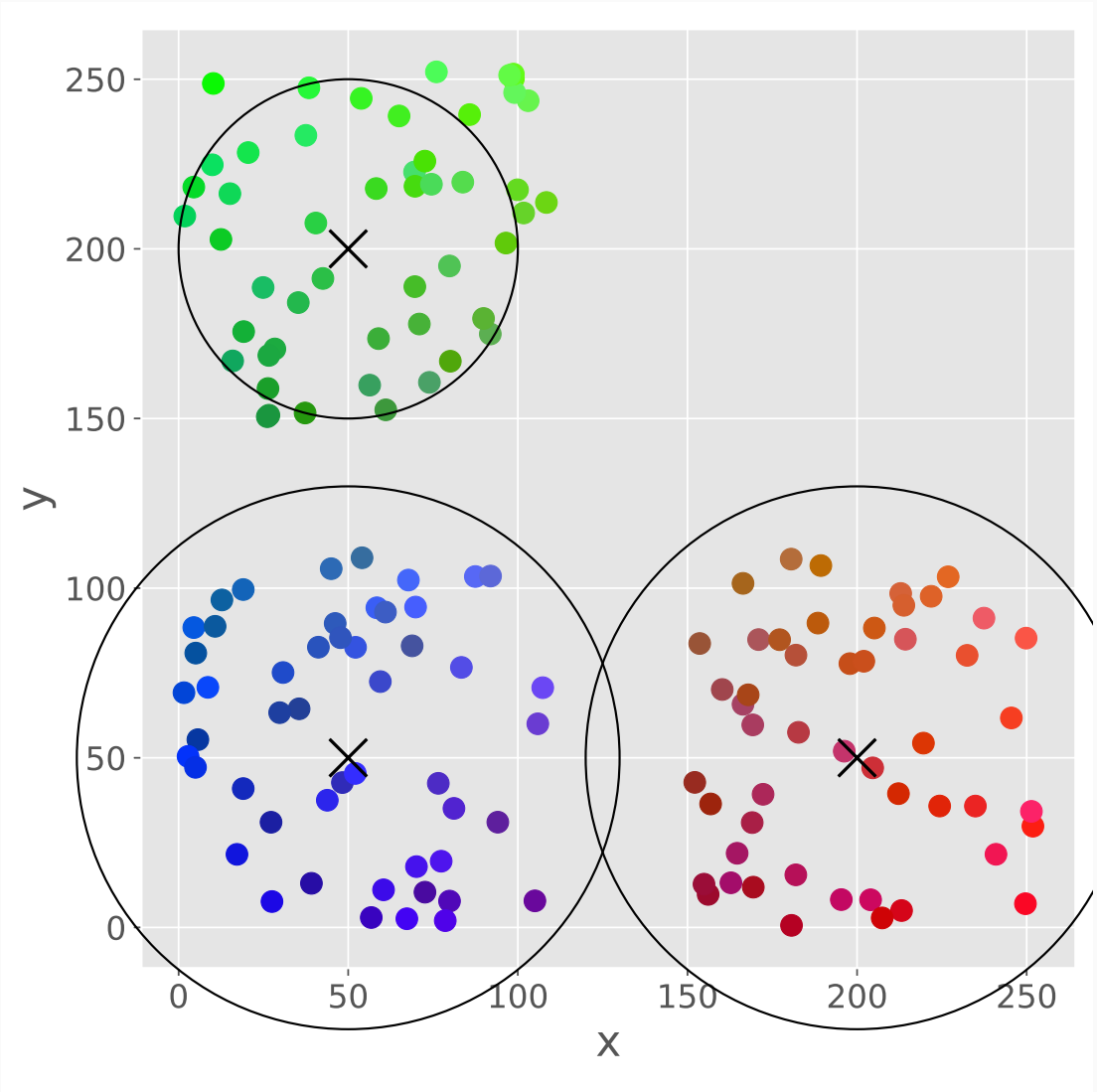
Repetimos esse procedimento...



UFABC

k-Means

...até os centros ou os grupos não se alterarem mais.



UFABC

k-Means - Algoritmo

```
def kmeans(X, k, it=100):
    centers = [X[i,:] for i in np.random.choice(X.shape[0])]
    while it>0:
        centers = emStep(X, centers)
    return estStep(X, centers)

def emStep(X, centers):
    return maxStep( X, estStep(X, centers) )
```

UFABC

k-Means - Algoritmo

```
def estStep(X, centers):
    return np.array([np.argmin(distances(x_i, centers))
                     for x_i in X])

def maxStep(X, labels):
    return [X[labels == k].mean(axis = 0)
            for k in np.unique(labels)]
```

UFABC

k-Means - Algoritmo

```
def distances(x, centers):
    return [dist(x, center_j) for center_j in centers]
```

UFABC

k-Means - Scikit-Learn

```
from sklearn.cluster import KMeans

model = KMeans(n_clusters=k)
model.fit(X)
clusters = model.labels_
```

UFABC

Próxima Aula

Na próxima aula aprenderemos sobre os algoritmos de agrupamento Espectral e Hierárquico.

UFABC