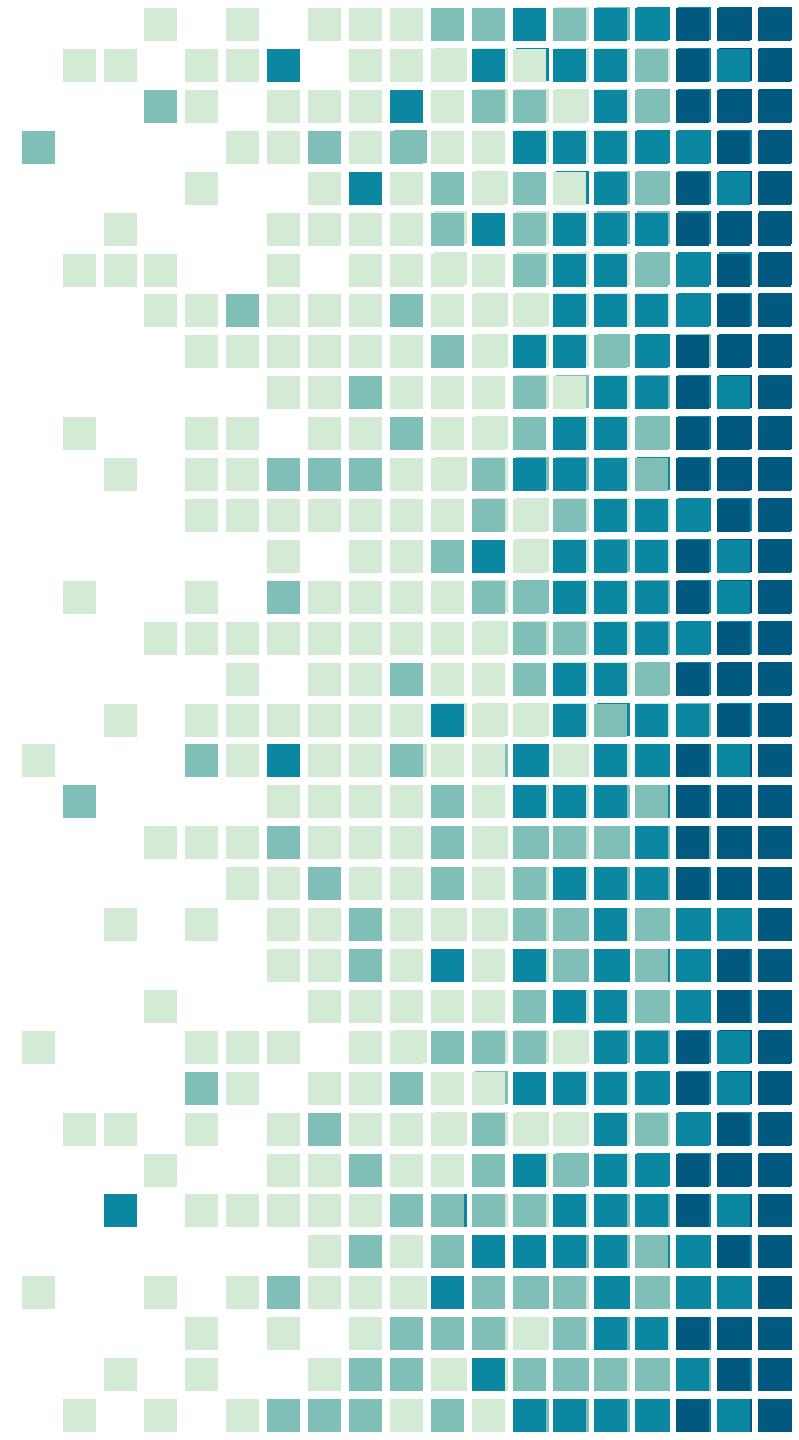


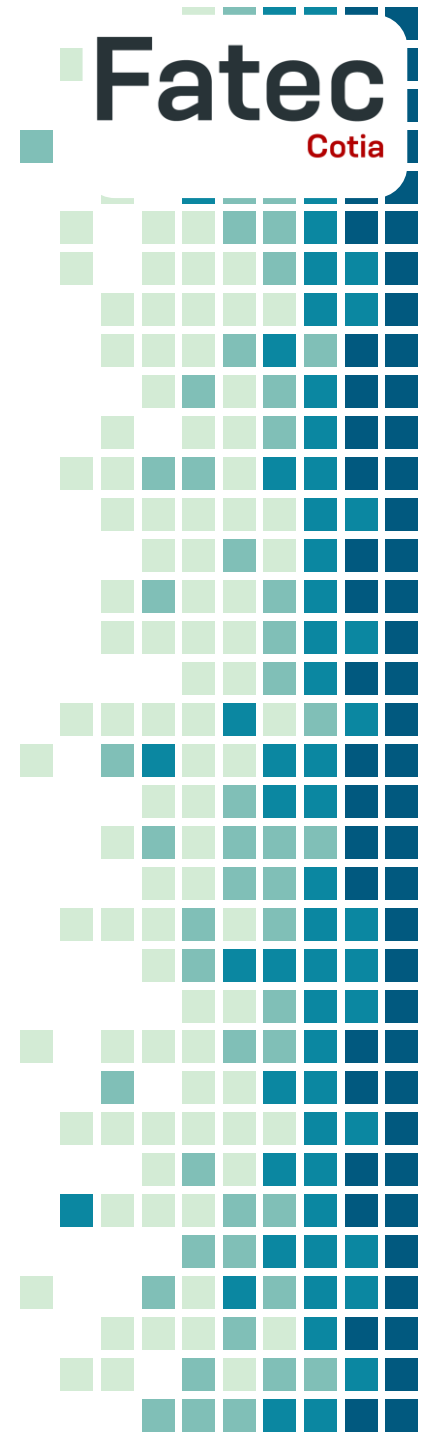


**SÃO PAULO**  
**GOVERNO DO ESTADO**

# Fatec

**Cotia**





# Disciplina: Mineração de Dados

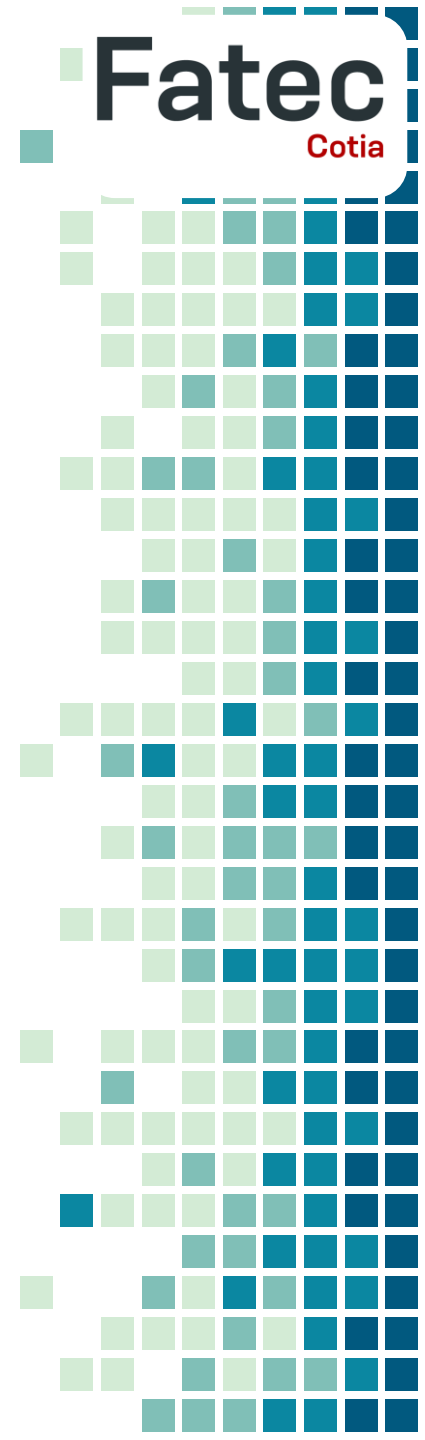
## Aula 13 - K-means

Curso: DSM | Desenvolvimento de Software Multiplataforma

Turma: 6º ciclo | 2024 | quinta-feira – noturno

Professor: Jeferson - Email: [jeferson.dias5@fatec.sp.gov.br](mailto:jeferson.dias5@fatec.sp.gov.br)

# K-MEANS



# K-MEANS



► K-means ou K-vizinhos mais próximos.

► Bibliografia:

- Data Science do Zero.
- Capítulo 12

# K-MEANS

- ▶ Os vizinhos mais próximos é um dos modelos preditivos mais simples que existe.
- ▶ Ele não possui premissas matemáticas e não requer nenhum tipo de maquinário pesado.



► Ele apenas requer:

- Uma noção de distância.
- Uma premissa de que pontos que estão perto um do outro são similares.



- Os vizinhos mais próximos, por outro lado, rejeitam muitas informações conscientemente, uma vez que a previsão para cada ponto novo depende somente de alguns pontos mais próximos.

# K-MEANS

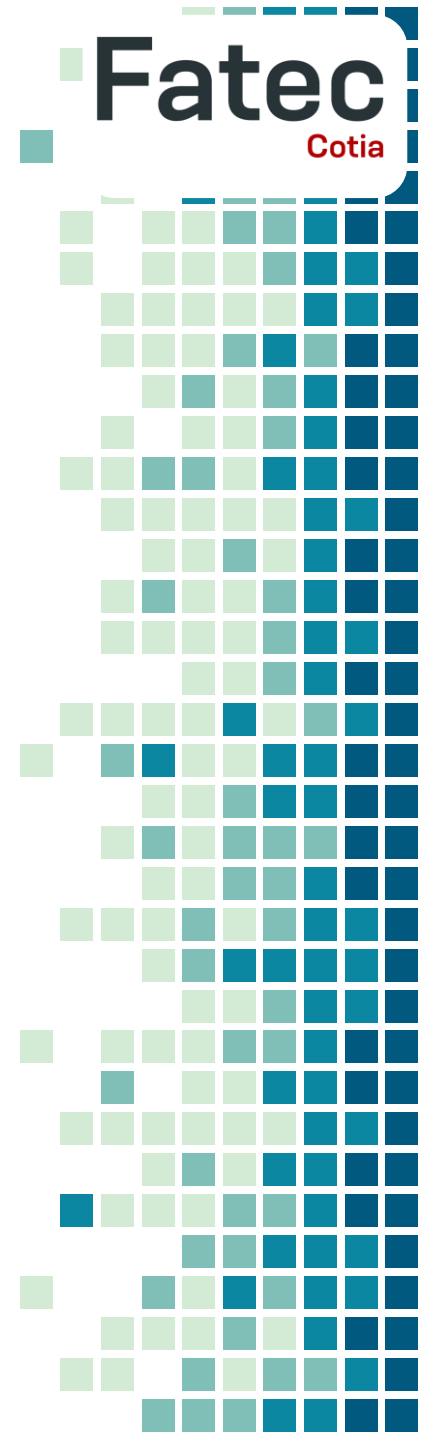


- Em uma situação geral, temos alguns pontos de dados e um conjunto de rótulos correspondentes.



- ▶ K-significa clustering exige que selecionemos K, o número de clusters em que queremos agrupar os dados.
- ▶ O método nos permite representar graficamente a inércia (uma métrica baseada na distância) e visualizar o ponto em que ela começa a diminuir linearmente.

# EXEMPLO 1



# EXEMPLO 1



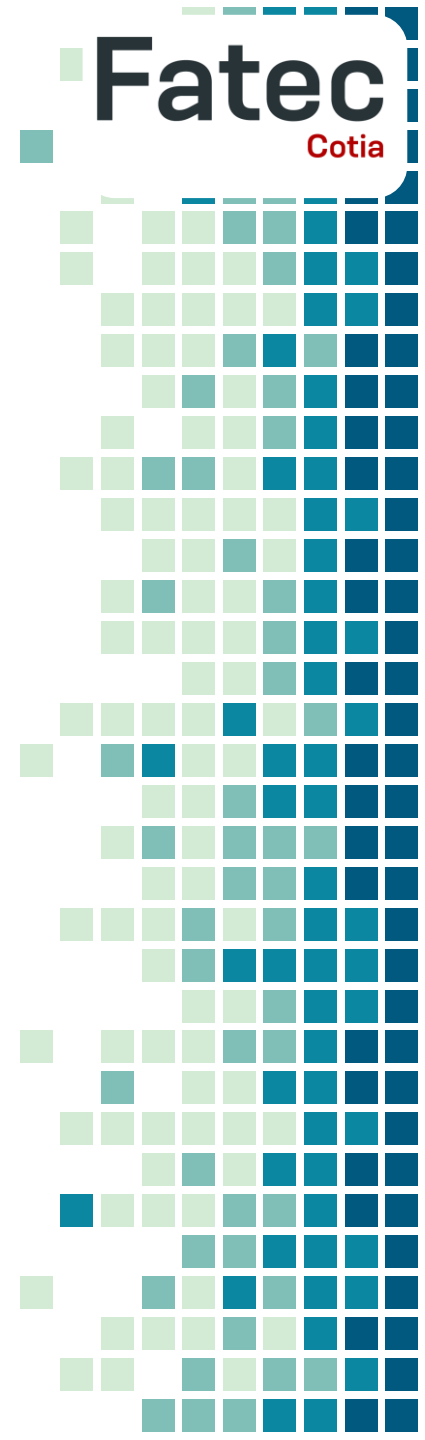
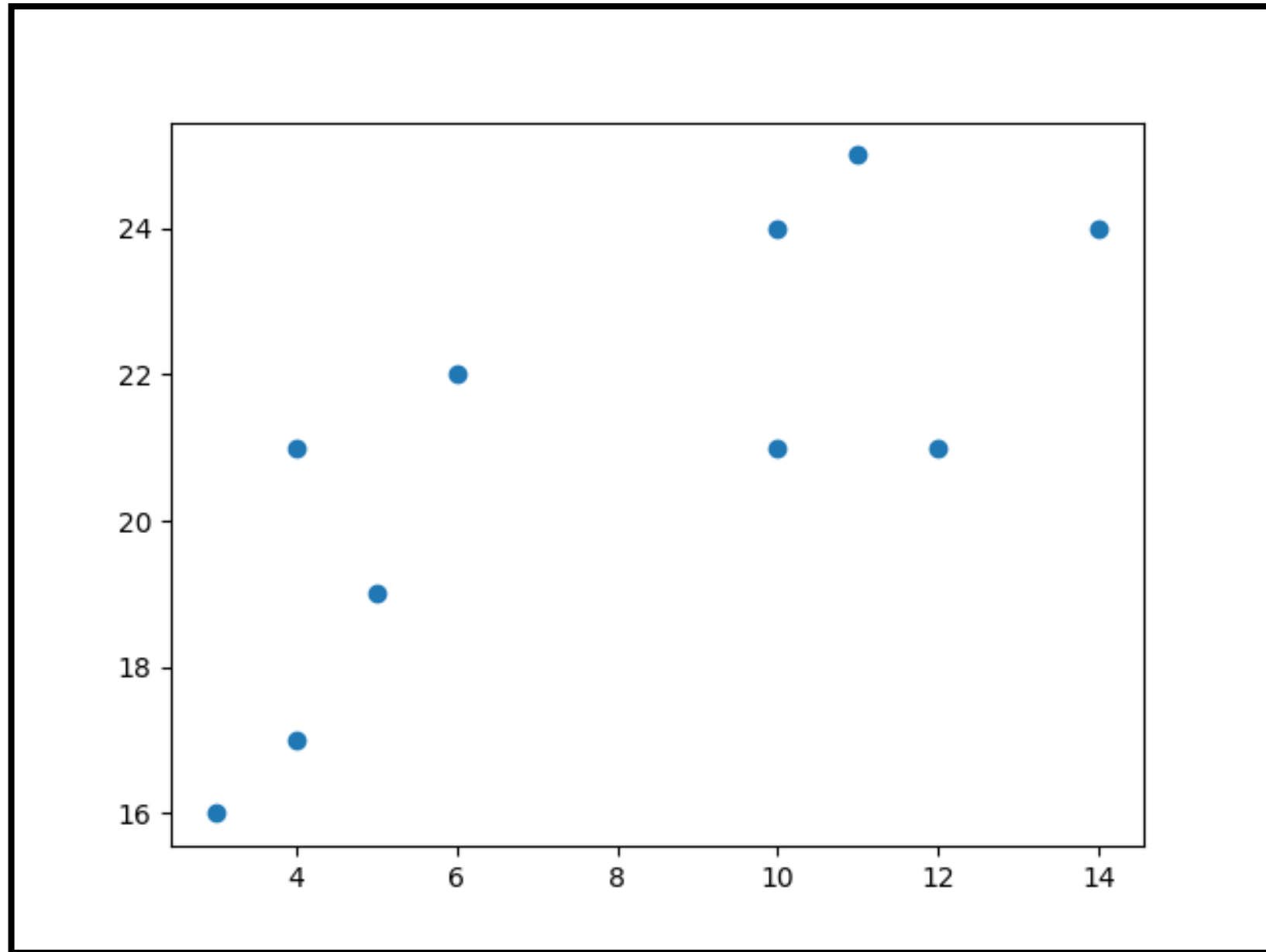
► Este exemplo simples, mostra os pontos de um gráfico k-means.

# EXEMPLO 1

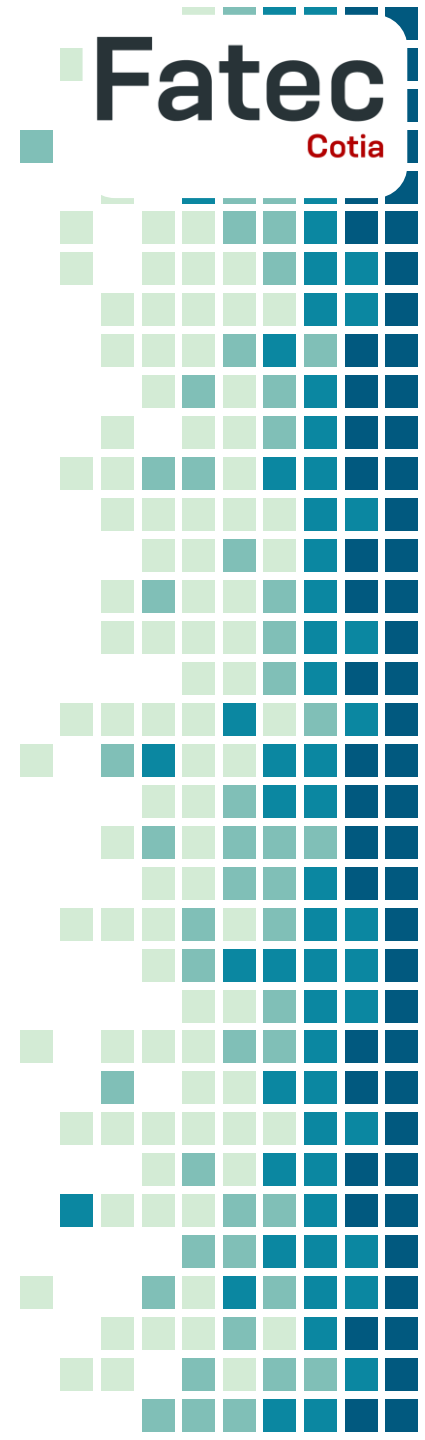
Exemplo1.py > ...

```
1  # um exemplo visual para entendimento do K-means
2
3  import matplotlib.pyplot as plt
4
5  # vetores
6  x = [4, 5, 10, 4, 3, 11, 14 , 6, 10, 12]
7  y = [21, 19, 24, 17, 16, 25, 24, 22, 21, 21]
8
9  # criação do gráfico
10 plt.scatter(x, y)
11 plt.show()
12
13
```

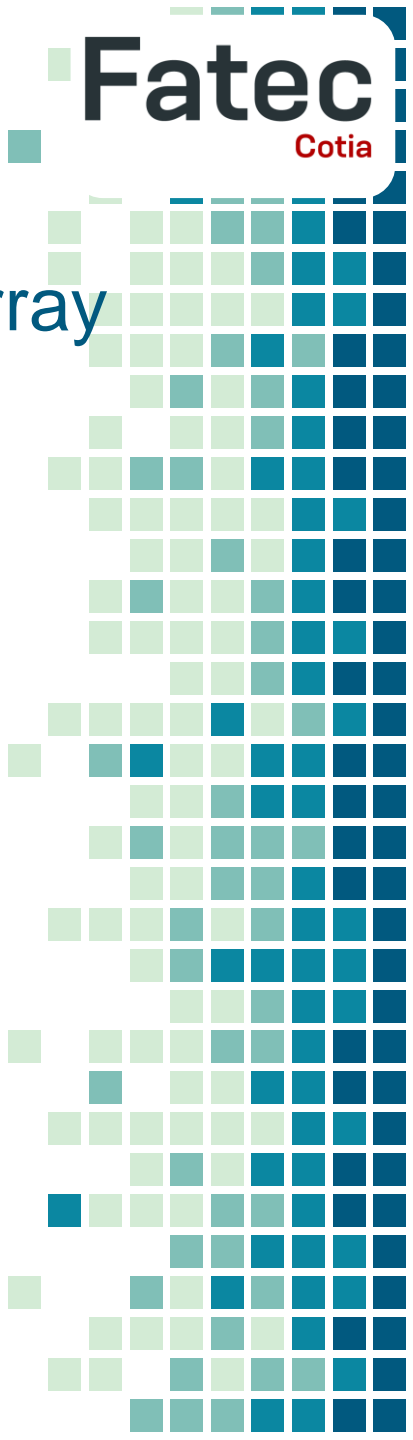
# EXEMPLO 1



# EXEMPLO 2



# EXEMPLO 2

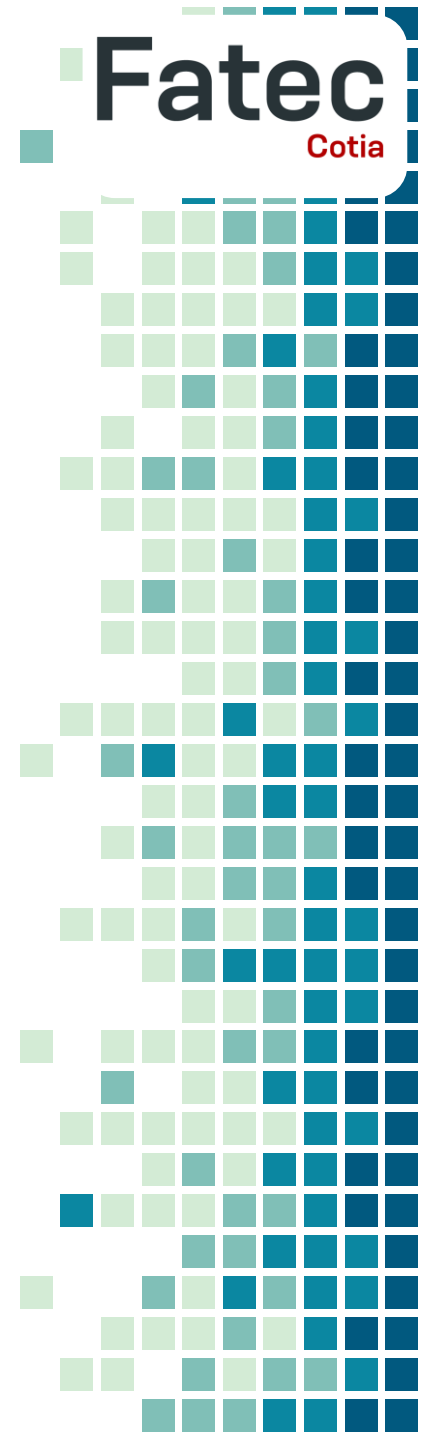


- ▶ Este exemplo vai gerar um cluster em 2D, baseado em um array aleatório.
- ▶ Estaremos usando uma biblioteca no python para Kmeans.

# EXEMPLO 2

► Instale a biblioteca:

```
pip install numpy matplotlib scikit-learn
```





# EXEMPLO 2



```
Exemplo2.py > ...  
1  import numpy as np  
2  import matplotlib.pyplot as plt  
3  from sklearn.cluster import KMeans  
4
```

# EXEMPLO 2

```
4
5 # Gerando dados fictícios em 2D
6 X = np.array([
7     [1, 2], [1.5, 1.8], [5, 8], [8, 8],
8     [1, 0.6], [9, 11], [8, 2], [10, 2],
9     [9, 3]
10 ])
11
```

# EXEMPLO 2

```
11
12 # Aplicando K-means com 3 clusters
13 kmeans = KMeans(n_clusters=3)
14 kmeans.fit(X)
15
16 # Obtendo os rótulos dos clusters
17 labels = kmeans.labels_
18
```

```
18
19 # Obtendo os centros dos clusters
20 # Extraímos as coordenadas dos centros dos clusters.
21 centers = kmeans.cluster_centers_
22
23 # Visualizando os clusters
24 colors = ['r', 'g', 'b']
25 for i in range(len(X)):
26     plt.scatter(X[i][0], X[i][1], c=colors[labels[i]], s=30)
27
```

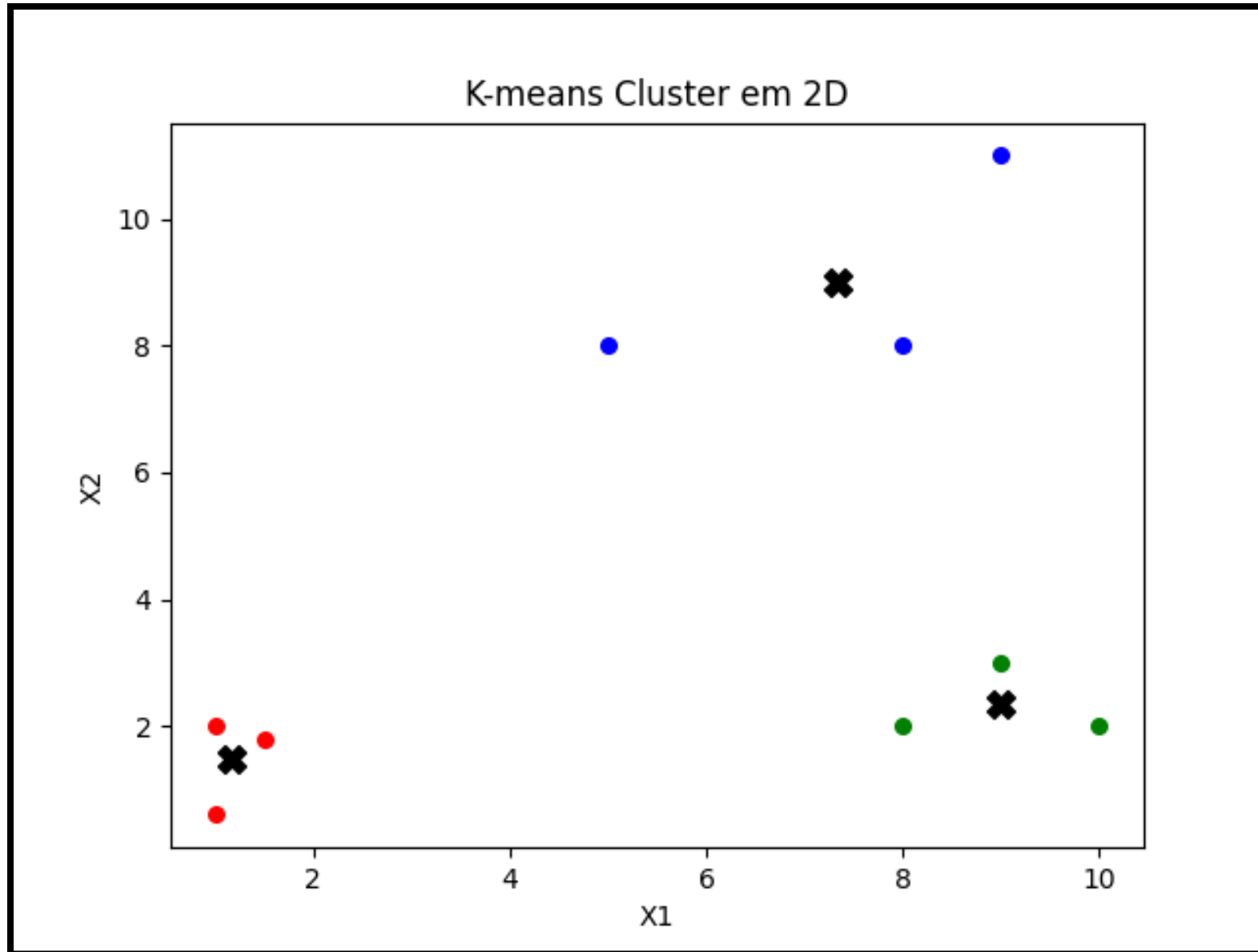
# EXEMPLO 2

```
27
28 # Visualizando os centros dos clusters
29 # Usamos plt.scatter para visualizar os pontos coloridos
30 # por cluster e os centros em preto.
31 plt.scatter(centers[:, 0], centers[:, 1], c='black', marker='X', s=100)
32 plt.title('K-means Cluster em 2D')
33 plt.xlabel('X1')
34 plt.ylabel('X2')
35 plt.show()
36
```

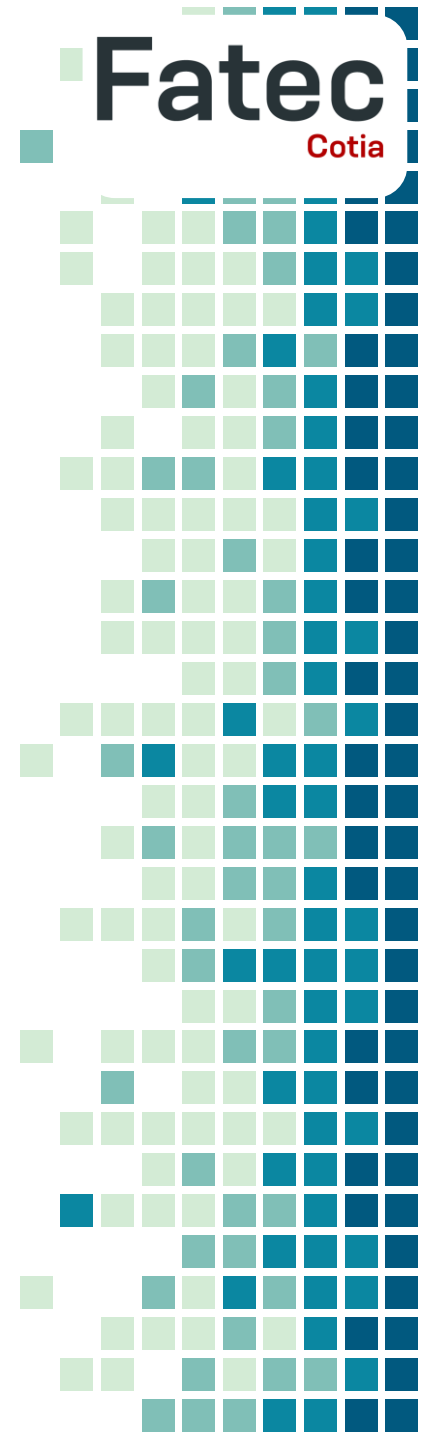
# EXEMPLO 2



► Saída:



# EXEMPLO 3



# EXEMPLO 3

- ▶ Mesmo exemplo anterior, mas usando gráfico 3D.





# EXEMPLO 3



```
Python Exemplo3.py > ...  
1  import numpy as np  
2  import matplotlib.pyplot as plt  
3  from mpl_toolkits.mplot3d import Axes3D  
4  from sklearn.cluster import KMeans  
5
```

# EXEMPLO 3

```
5
6 # Gerando dados fictícios em 3D
7 X = np.array([
8     [1, 2, 3], [1.5, 1.8, 2.5], [5, 8, 7], [8, 8, 9],
9     [1, 0.6, 1.2], [9, 11, 10], [8, 2, 3], [10, 2, 6],
10    [9, 3, 5]
11 ])
12
```

# EXEMPLO 3

```
12
13 # Aplicando K-means com 3 clusters
14 kmeans = KMeans(n_clusters=3)
15 kmeans.fit(X)
16
17 # Obtendo os rótulos dos clusters
18 labels = kmeans.labels_
19
20 # Obtendo os centros dos clusters
21 centers = kmeans.cluster_centers_
22
```

# EXEMPLO 3

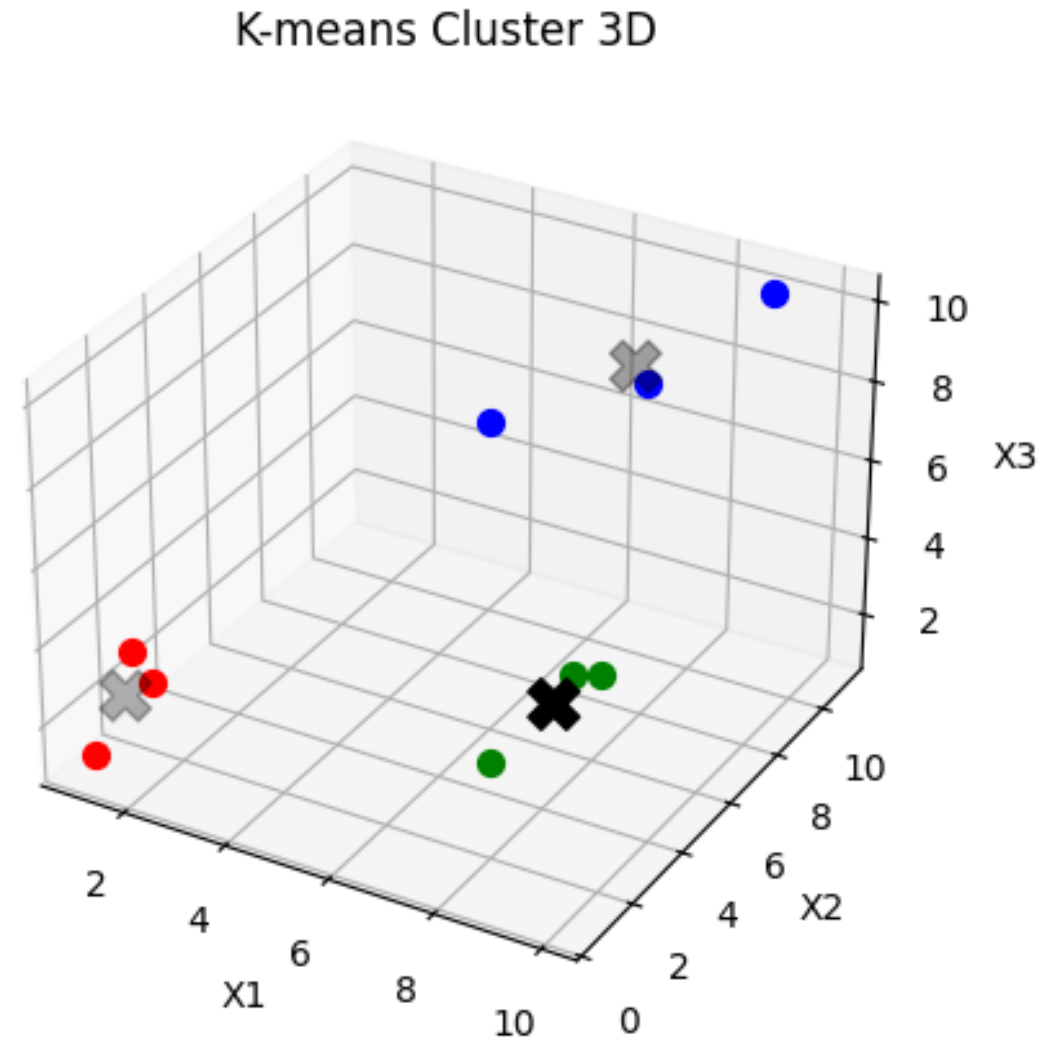
```
22
23 # Visualizando os clusters em 3D
24 fig = plt.figure()
25 ax = fig.add_subplot(111, projection='3d')
26
27 colors = ['r', 'g', 'b']
28 for i in range(len(X)):
29     ax.scatter(X[i][0], X[i][1], X[i][2], c=colors[labels[i]], s=50)
30
```

# EXEMPLO 3

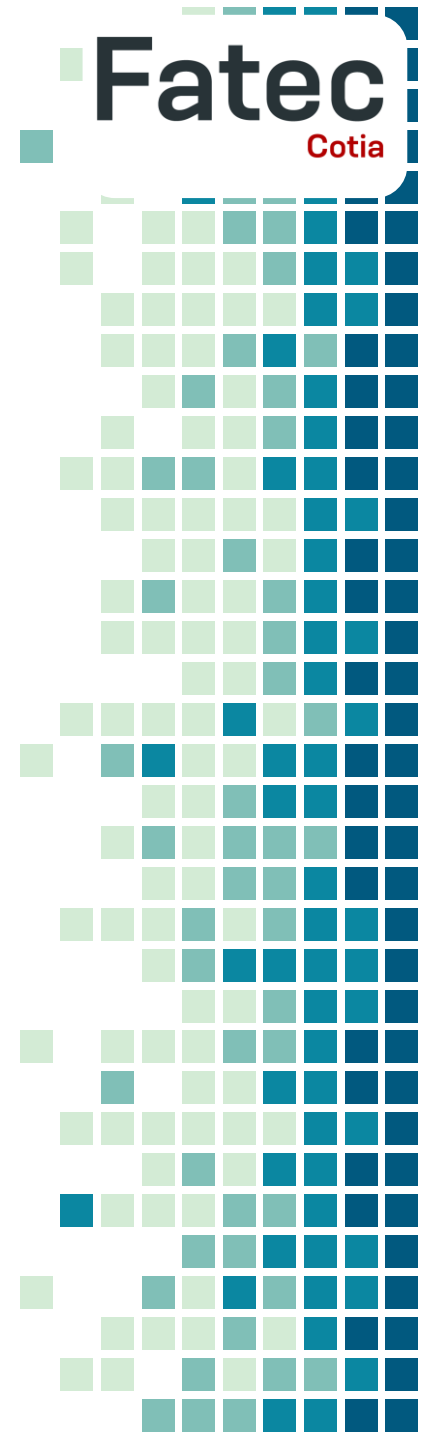
```
30
31 # Visualizando os centros dos clusters
32 ax.scatter(centers[:, 0], centers[:, 1], centers[:, 2], c='black', marker='X', s=200)
33 ax.set_title('K-means Cluster 3D')
34 ax.set_xlabel('X1')
35 ax.set_ylabel('X2')
36 ax.set_zlabel('X3')
37 plt.show()
38
```

# EXEMPLO 3

► Saída:



# EXEMPLO 4



# EXEMPLO 4

► Para este exemplo, preciso entender:





# EXEMPLO 4

- **Definir o Número de Clusters (K):** Primeiramente, é necessário decidir quantos grupos você deseja criar. Vamos supor que você queira identificar três perfis de clientes. Logo,  $K = 3$ .

# EXEMPLO 4

- **Inicializar os Centroides:** O próximo passo é escolher pontos iniciais que representarão os centros dos clusters, chamados de “centroides”. Esses pontos são escolhidos aleatoriamente no início.

# EXEMPLO 4

- ▶ **Como Escolher o Valor de K?**
- ▶ Escolher o valor correto para K pode ser um desafio, pois um valor inadequado pode levar a grupos pouco úteis ou não significativos.
- ▶ Para auxiliar nesta escolha, utilizamos o método do “Cotovelo” (Elbow Method).

# EXEMPLO 4

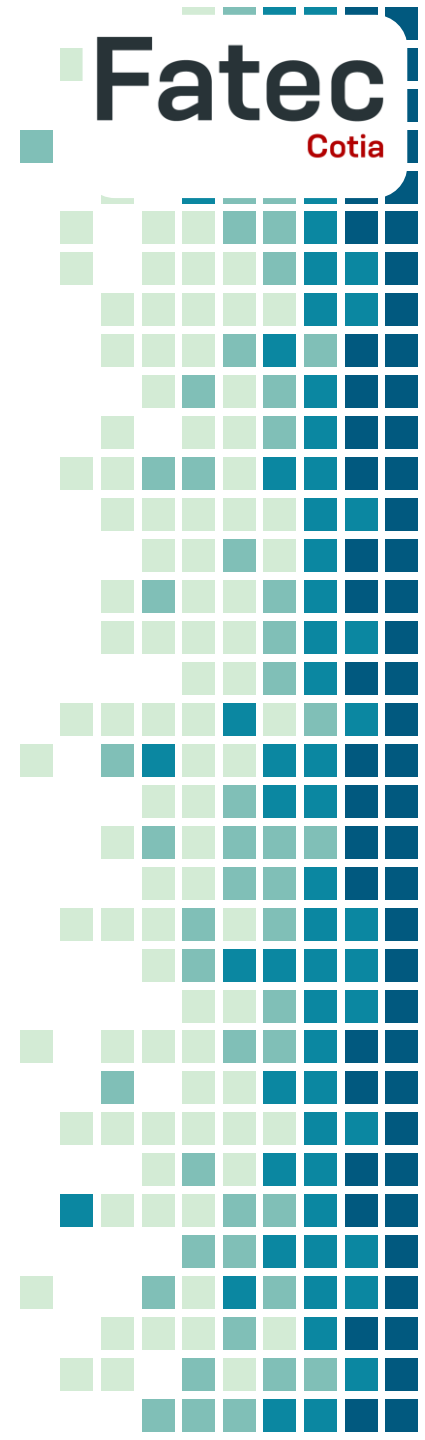
Exemplo4\_A.py > ...

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  # Gerando dados fictícios em 2D
5  X = np.array([
6      [1, 2], [1.5, 1.8], [5, 8], [8, 8],
7      [1, 0.6], [9, 11], [8, 2], [10, 2],
8      [9, 3], [6, 7], [3, 3], [7, 9],
9      [2, 5], [3.5, 2.8], [4, 2], [8, 6],
10 ])
11
```

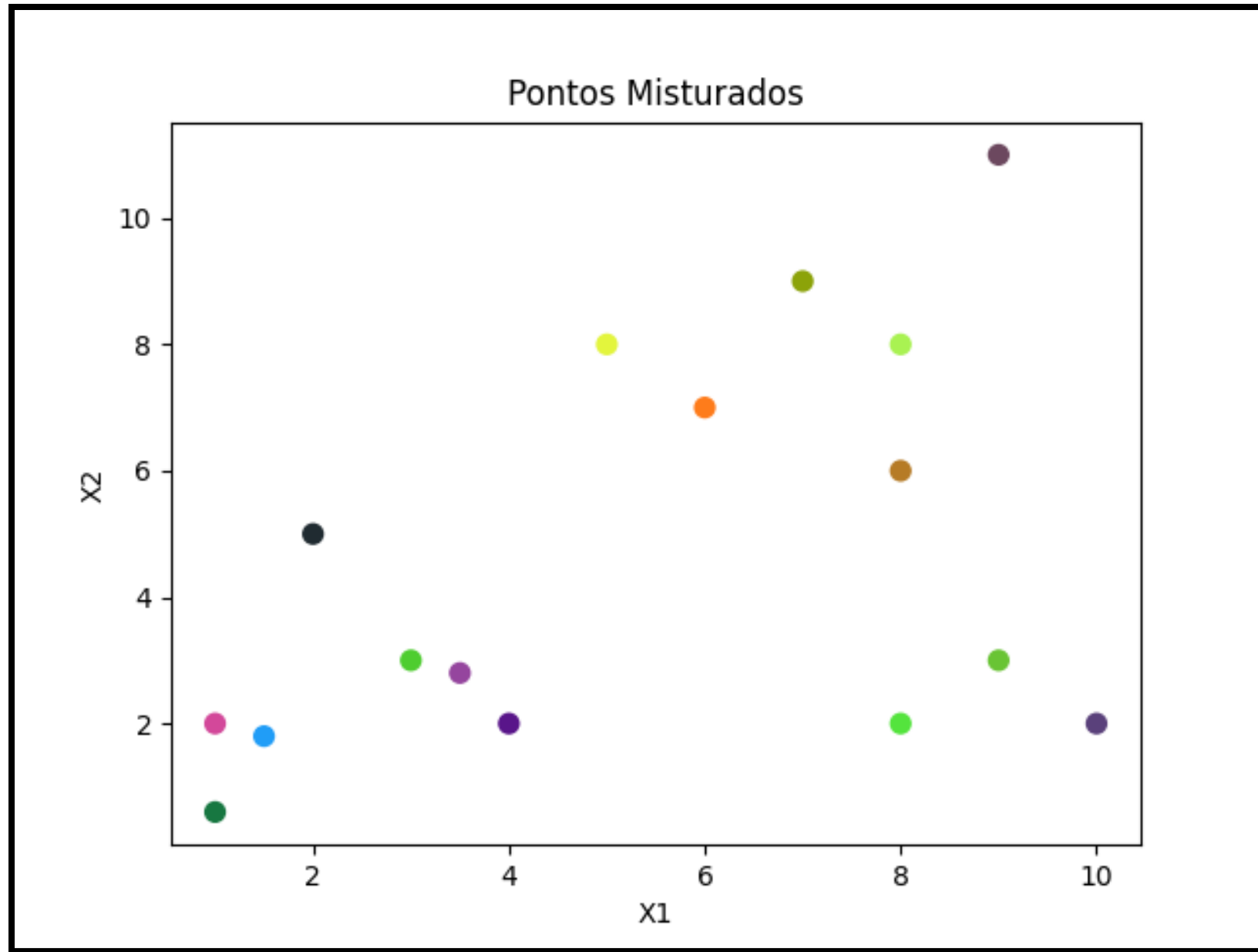
# EXEMPLO 4

```
11
12 # Gerando cores aleatórias para os pontos
13 colors = np.random.rand(len(X), 3)
14
15 # Visualizando os pontos misturados
16 plt.scatter(X[:, 0], X[:, 1], c=colors, s=50)
17 plt.title('Pontos Misturados')
18 plt.xlabel('X1')
19 plt.ylabel('X2')
20 plt.show()
21
```

# EXEMPLO 4



► Saída:



# EXEMPLO 4

- Segunda parte e criar centroides, com isso vai mudar a posição dos pontos.

# EXEMPLO 4

```
Exemplo4_B.py > ...  
1  import numpy as np  
2  import matplotlib.pyplot as plt  
3  from sklearn.cluster import KMeans  
4  
5  # Gerando dados fictícios em 2D  
6  # (mesmo conjunto de dados do programa anterior)  
7  X = np.array([  
8      [1, 2], [1.5, 1.8], [5, 8], [8, 8],  
9      [1, 0.6], [9, 11], [8, 2], [10, 2],  
10     [9, 3], [6, 7], [3, 3], [7, 9],  
11     [2, 5], [3.5, 2.8], [4, 2], [8, 6],  
12 ])  
13
```



# EXEMPLO 4

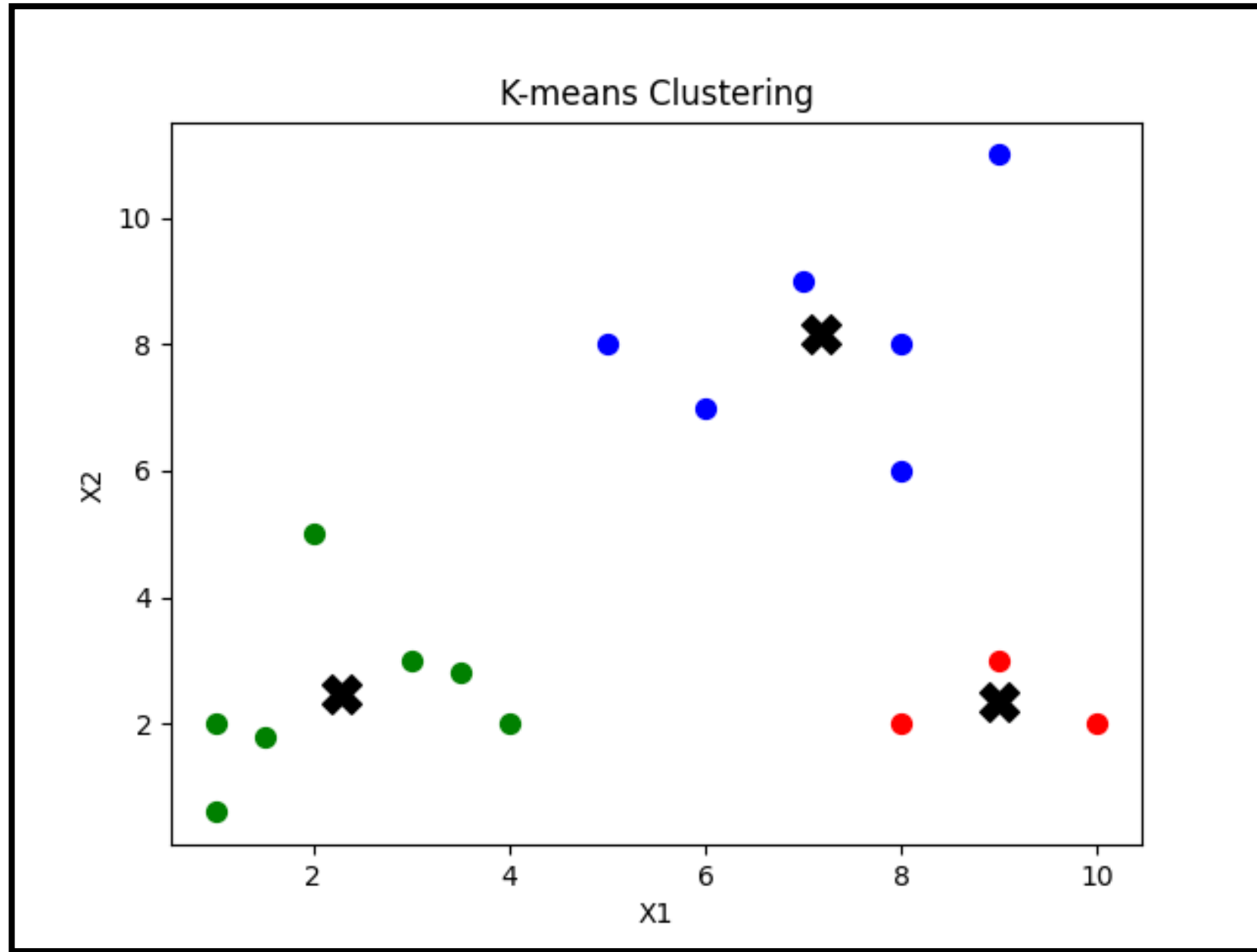
```
13
14 # Aplicando K-means com 3 clusters
15 kmeans = KMeans(n_clusters=3)
16 kmeans.fit(X)
17
18 # Obtendo os rótulos dos clusters
19 labels = kmeans.labels_
20
21 # Obtendo os centros dos clusters
22 centers = kmeans.cluster_centers_
23
```

# EXEMPLO 4

```
23
24 # Visualizando os clusters
25 colors = ['r', 'g', 'b']
26 for i in range(len(X)):
27     plt.scatter(X[i][0], X[i][1], c=colors[labels[i]], s=50)
28
29 # Visualizando os centros dos clusters
30 plt.scatter(centers[:, 0], centers[:, 1], c='black', marker='X', s=200)
31 plt.title('K-means Clustering')
32 plt.xlabel('X1')
33 plt.ylabel('X2')
34 plt.show()
35
```

# EXEMPLO 4

► Saída:



# EXEMPLO 5




# EXEMPLO 5

- ▶ Este exemplo, vai mostrar os pontos de maior populações por estado do Estados Unidos da América.
- ▶ Precisamos desta biblioteca:

```
pip install numpy matplotlib scikit-learn basemap
```

# EXEMPLO 5

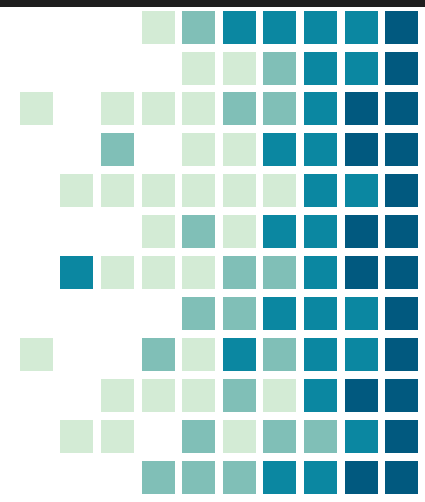
```
Exemplo5.py > ...  
1  import numpy as np  
2  import matplotlib.pyplot as plt  
3  from sklearn.cluster import KMeans  
4  from mpl_toolkits.basemap import Basemap  
5
```



# EXEMPLO 5



```
5 |
6 # Gerando pontos aleatórios representando populações em coordenadas dos EUA
7 np.random.seed(42) # Para reprodutibilidade
8 num_points = 100
9 lats = np.random.uniform(25, 50, num_points) # Latitude entre 25 e 50
10 lons = np.random.uniform(-125, -66, num_points) # Longitude entre -125 e -66
11 populations = np.random.randint(1, 1000, num_points) # População entre 1 e 1000
12
```



# EXEMPLO 5



```
12
13     # Gerando dados para K-means
14     X = np.column_stack((lons, lats))
15
16     # Aplicando K-means com 5 clusters
17     kmeans = KMeans(n_clusters=5)
18     kmeans.fit(X)
19
20     # Obtendo os rótulos dos clusters
21     labels = kmeans.labels_
22
23     # Obtendo os centros dos clusters
24     centers = kmeans.cluster_centers_
25
```



# EXEMPLO 5

```
25
26 # Visualizando os clusters no mapa
27 plt.figure(figsize=(12, 8))
28 m = Basemap(projection='merc',
29             llcrnrlat=24,
30             urcrnrlat=50,
31             llcrnrlon=-125,
32             urcrnrlon=-66,
33             lat_ts=20,
34             resolution='i')
35 m.drawcoastlines()
36 m.drawcountries()
37 m.drawstates()
38 m.drawmapboundary(fill_color='aqua')
39 m.fillcontinents(color='lightgreen', lake_color='aqua')
40
```

# EXEMPLO 5

```
40
41 # Convertendo coordenadas para o mapa
42 x, y = m(lons, lats)
43 x_center, y_center = m(centers[:, 0], centers[:, 1])
44
45 colors = ['r', 'g', 'b', 'c', 'm']
46 for i in range(num_points):
47     m.scatter(x[i], y[i], s=populations[i] / 10, c=colors[labels[i]], alpha=0.6)
48
```

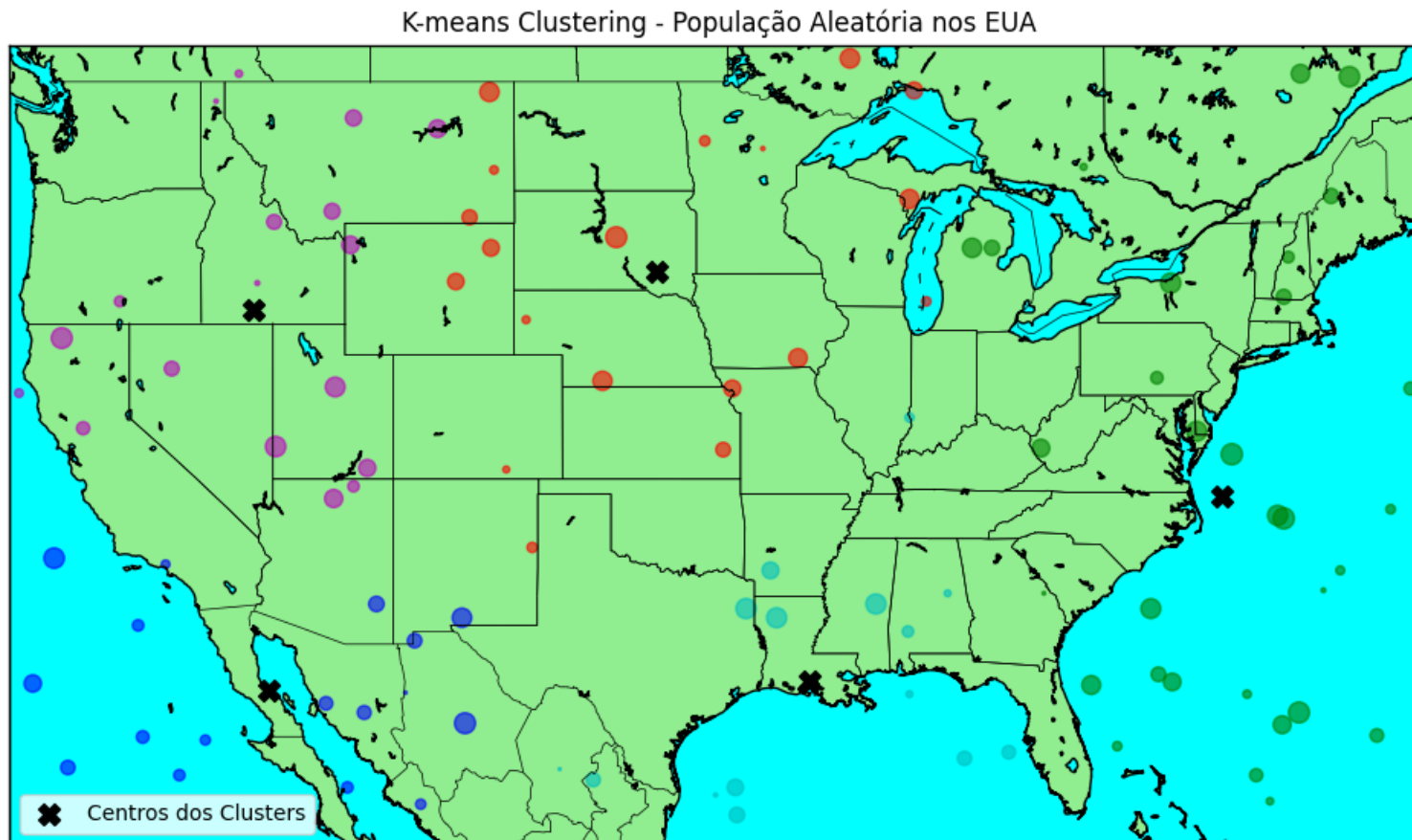
# EXEMPLO 5



```
48
49 # Visualizando os centros dos clusters
50 m.scatter(x_center, y_center, c='black', marker='X', s=100, label='Centros dos Clusters')
51 plt.legend(loc='lower left')
52 plt.title('K-means Clustering - População Aleatória nos EUA')
53 plt.show()
54 |
```

# EXEMPLO 5

► Saída:





# Obrigado

Qualquer dúvida entrar em contato via e-mail:

E-mail [jeferson.dias5@fatec.sp.gov.br](mailto:jeferson.dias5@fatec.sp.gov.br)