

## TP : Clustering avec K-Means - Correction

1. Importer les librairies nécessaires

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
```

2. Création du jeu de données (6 points)

```
data = np.array([
    [1, 2], [1, 4], [2, 3], [5, 8], [6, 9], [5, 7]
])

df = pd.DataFrame(data, columns=['X', 'Y'])
```

3. Tracer le nuage de points avant l'application de K-Means

```
plt.figure(figsize=(8,6))
plt.scatter(df['X'], df['Y'], color='blue', label="Données initiales")
plt.xlabel("X")
plt.ylabel("Y")
plt.title("Nuage de points avant l'application de K-Means")
plt.legend()
plt.grid(True)
plt.show()
```

4. Définition des centroïdes initiaux (1er et 4ème point)

```
initial_centroids = np.array([
    df.iloc[0], # (1,2)
    df.iloc[3]  # (5,8)
])
```

## 5. Application de K-Means avec ces centroïdes initiaux

```
kmeans = KMeans(n_clusters=2, init=initial_centroids, max_iter=100, n_init=1, random_state=42)
kmeans.fit(df[['X', 'Y']])

# Nombre d'itérations avant la convergence
iterations = kmeans.n_iter_

# Récupération des centroïdes finaux
final_centroids = kmeans.cluster_centers_
```

## 6. Affichage des résultats

```
plt.figure(figsize=(8,6))
plt.scatter(df['X'], df['Y'], c=kmeans.labels_, cmap='viridis', label="Points")
plt.scatter(initial_centroids[:, 0], initial_centroids[:, 1], c='blue', marker='D', s=200, label="Centroïdes initiaux")
plt.scatter(final_centroids[:, 0], final_centroids[:, 1], c='red', marker='X', s=200, label="Centroïdes finaux")
plt.xlabel("X")
plt.ylabel("Y")
plt.title(f"K-Means avec centroïdes donnés (Converge en {iterations} itérations)")
plt.legend()
plt.grid(True)
plt.show()
```

## 7. Affichage des informations

```
print(f"Centres des clusters après convergence :\n{final_centroids}")
print(f"Nombre d'itérations avant la convergence : {iterations}")
```