corrigé

L'algorithme des k-moyennes

1. L'algorithme des k-moyennes

Question 1.

```
def barycentre(data):
    x, y = 0, 0
    for p in data:
        x += p[0]
        y += p[1]
    return [x / len(data), y / len(data)]
```

Question 2.

```
def dist(p, q):
    return (p[0] - q[0])**2 + (p[1] - q[1])**2
```

Question 3.

```
def plusProche(p, mu):
    dmin = float('inf')
    for i in range(len(mu)):
        if dist(p, mu[i]) < dmin:
            dmin, imin = dist(p, mu[i]), i
    return imin
```

Question 4.

```
def kmeans(data, k):
    mu = sample(data, k)
    while True:
        clusters = [[] for _ in range(k)]
        for p in data:
            i = plusProche(p, mu)
            clusters[i].append(p)
        newmu = [barycentre(clusters[i]) for i in range(k)]
        if mu == newmu:
            return clusters
        mu = newmu
```

On observera sur la figure 1 que le partitionnement obtenu n'est pas nécessairement optimal.

Question 5.

```
def inertie(clusters):
    m = 0
    for data in clusters:
        mu = barycentre(data)
        for p in data:
            m += dist(p, mu)
    return m
```

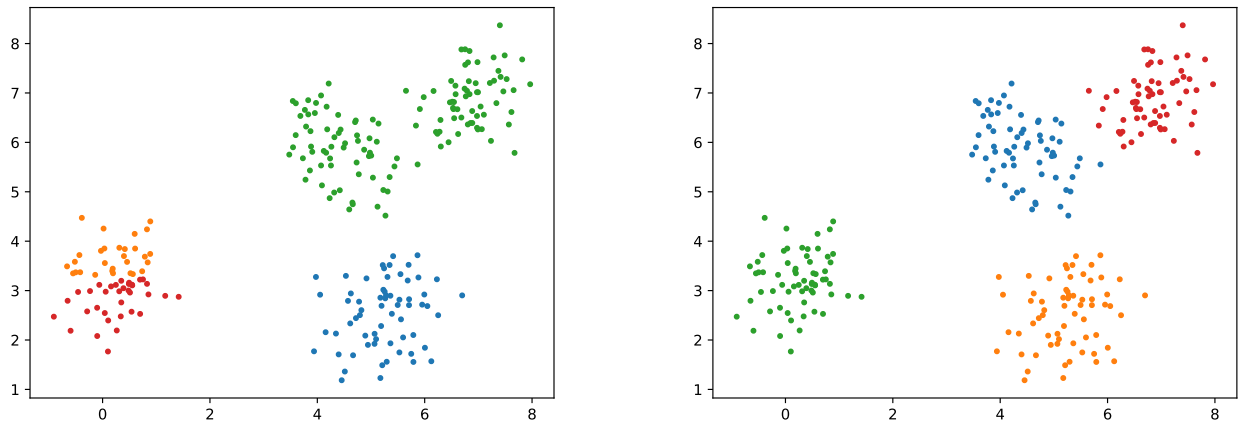

FIGURE 1 – Deux partitionnements d'un même ensemble, d'inerties respectives 272,06 et 185.88

Question 6.

```
def bestKmeans(data, k):
    mini = float('inf')
    for _ in range(20):
        clusters = kmeans(data, k)
        inert = inertie(clusters)
        if inert < mini:
            mini, clustersmin = inert, clusters
    return clustersmin
```

2. Le choix de l'entier k

Question 7.

```
def moyenne(lst):
    return sum(lst) / len(lst)
```

Question 8.

```
def a(p, clusters):
    j = 0
    while p not in clusters[j]:
        j += 1
    return moyenne([dist(p, q) for q in clusters[j] if q != p])
```

```
def b(p, clusters):
    smin = float('inf')
    for clust in clusters:
        if p not in clust:
            smin = min(smin, moyenne([dist(p, q) for q in clust]))
    return smin
```

```python
def sil(p, clusters):
    ap, bp = a(p, clusters), b(p, clusters)
    return (bp - ap) / max(ap, bp)
```

Question 9.

```python
def silhouette(clusters):
    return moyenne([moyenne([sil(p, clusters) for p in c]) for c in clusters])
```

J'ai représenté figure 2 les valeurs de silhouette des partitions obtenues pour `data2` pour différentes valeurs de k. On constate que la valeur $k = 3$ est celle correspondant au facteur de silhouette maximal.

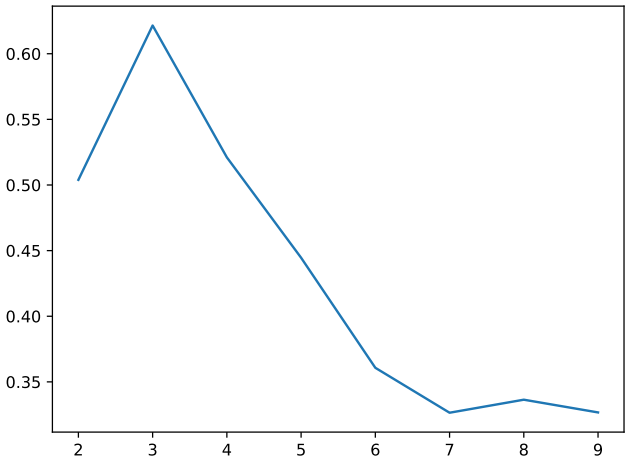

FIGURE 2 – Les valeurs des silhouettes du partitionnement de `data2` pour différentes valeurs de k