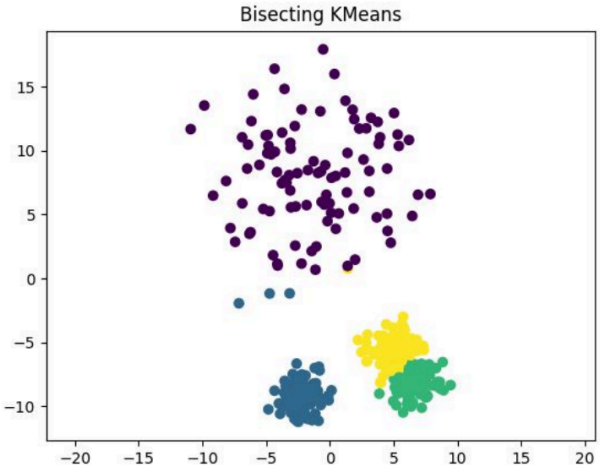
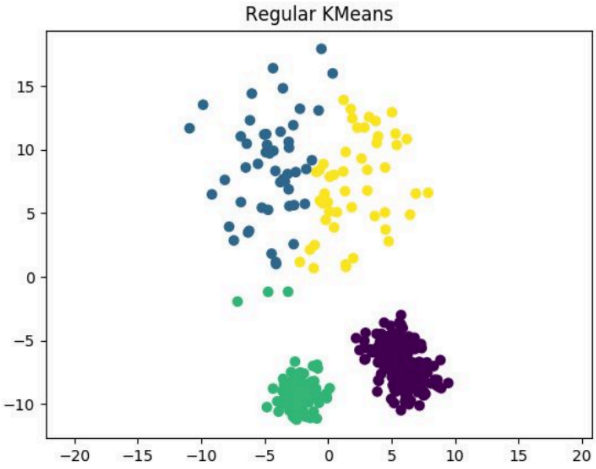
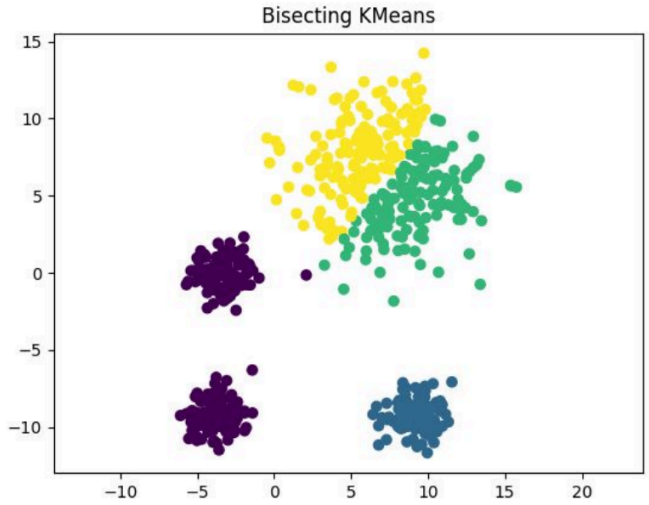
AS3 solutions

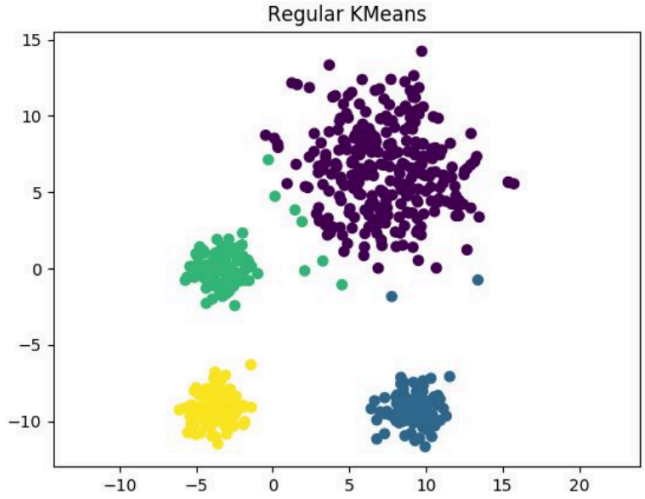
1A) This solution courtesy of John Verwolf. See attached code bisect_kmeans_sol.py
1B)





1C)


Bisecting KMeans
15
10
5
0
−5
−10
−10
−5
0
5
10
15
20

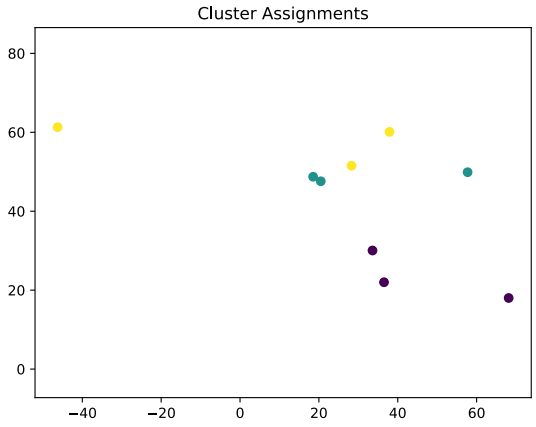

Regular KMeans
15
10
5
0
−5
−10
−10
−5
0
5
10
15
20

2) The two measures are not equivalent.

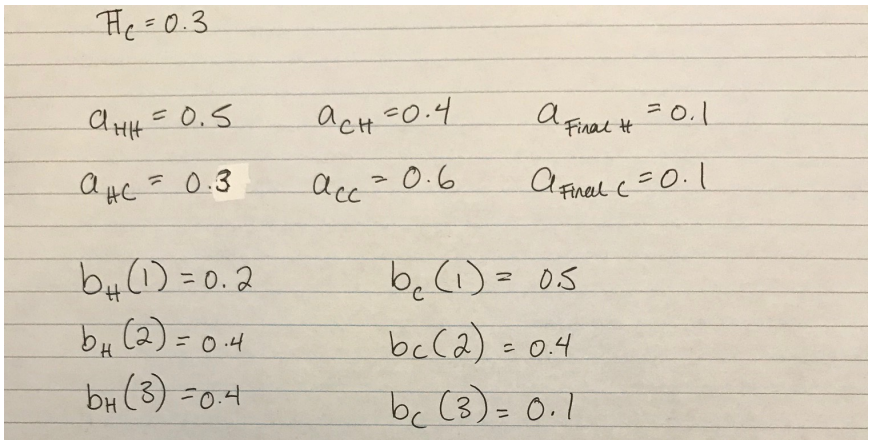
Here's an example dataset:
9 2D points
[[38.4340567   59.83877247]
 [44.35538753 37.39158448]
 [45.1730023   21.43245629]
 [31.48667678 49.85304981]
 [38.81901638 36.38664385]
 [80.5184922   65.10028981]
 [16.18189263 28.60428027]
 [37.74581653 18.59979592]
 [55.69119448 29.71023687]]
assignment of points to clusters:[2 1 0 1 1 2 2 0 0]

average distances
01 21.8570766195 *** min
02 41.1900999753
12 31.7785097322
average of centroid distances
01 19.6569268541
02 27.9576319157
12 12.0825406717 *** min

Average distances selects cluster 0 and 1 to merge, average of centroid distances selects clusters 1 and 2 to merge.



Q3) There is more than one right answer for this question, but here is one possible solution



State \ t	$O_1 = 3$ $t=1$	$O_2 = 1$ $t=2$	FINAL
H	0.28	→ 0.028	→ 0.0028
C	0.03	→ 0.056	→ 0.0056

Seq = 3, 1

$\delta_1(H) = \pi_H \, b_H(3) = 0.7 \times 0.4 = 0.28$

$\delta_1(C) = \pi_C \, b_C(3) = 0.3 \times 0.1 = 0.03$

$\delta_2(H) = \max\left[\delta_1(H)\, a_{HH}\, b_H(1), \delta_1(C)\, a_{HC}\, b_H(1)\right]$

$= \max\left[0.28 \times 0.5 \times 0.2, 0.03 \times 0.3 \times 0.2\right]$

$= \max\left[0.028, 0.0018\right]$

$\delta_2(C) = \max\left[\delta_1(H)\, a_{CH}\, b_C(1), \delta_1(C)\, a_{CC}\, b_C(1)\right]$

$= \max\left[0.28 \times 0.4 \times 0.5, 0.03 \times 0.6 \times 0.5\right]$

$= \max\left[0.056, 0.009\right]$

Sequence is H, C

State \ t	$O_1=1$ $t=1$	$O_2=1$ $t=2$	$O_3=3$ $t=3$	FINAL
H	0.15 → 0.015		0.003 →	0.0003
C	0.14 → 0.042		→ 0.00252	→ 0.00252

Seq : 1, 1, 3

$\delta_1(H) = \pi_H b_H(1) = 0.7 \times 0.2 = 0.14$

$\delta_1(C) = \pi_C b_C(1) = 0.3 \times 0.5 = 0.15$

$\delta_2(H) = \max\left[\delta_1(H) a_{HH} b_H(1), \delta_1(C) a_{HC} b_H(1)\right]$

$= \max\left[0.15 \times 0.5 \times 0.2, 0.14 \times 0.3 \times 0.2\right]$

$= \max\left[\underline{0.015}, 0.0084\right]$

$\delta_2(C) = \max\left[\delta_1(H) a_{CH} b_C(1), \delta_1(C) a_{CC} b_C(1)\right]$

$= \max\left[0.15 \times 0.4 \times 0.5, 0.14 \times 0.6 \times 0.5\right]$

$= \max\left[0.03, \underline{0.042}\right]$

$\delta_3(H) = \max\left[\delta_2(H) a_{HH} b_H(3), \delta_2(C) a_{HC} b_H(3)\right]$

$= \max\left[0.015 \times 0.5 \times 0.4, 0.042 \times 0.3 \times 0.4\right]$

$= \max\left[0.003, \underline{0.00504}\right]$

$$\delta_3(C) = \max\left[\delta_2(H)\, a_{CH}\, b_C(3),\ \delta_2(C)\, a_{CC}\, b_C(3)\right]$$

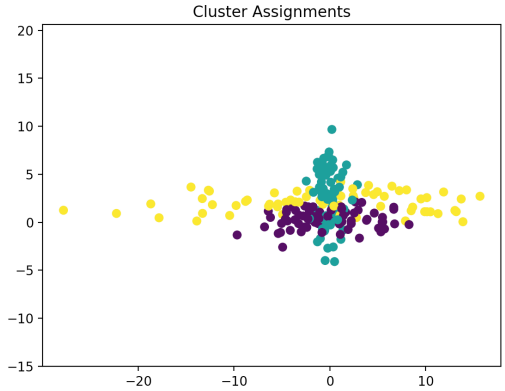
$$= \max\left[0.015 \times 0.4 \times 0.1,\ 0.042 \times 0.6 \times 0.1\right]$$

$$= \max\left[0.0006,\ \underline{0.00252}\right]$$

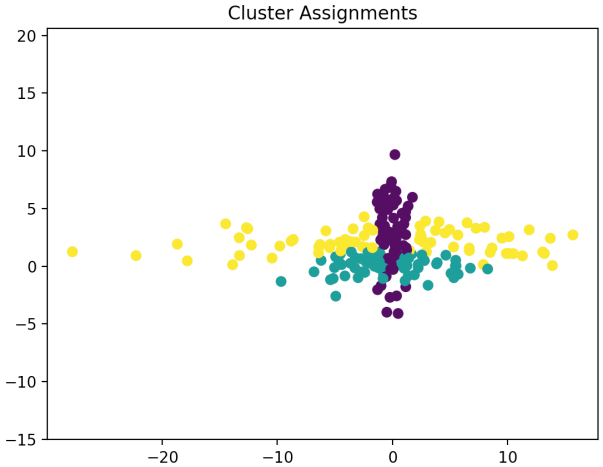
Final sequence is C, C, H

Q4A)
See em_sol.py

Q4B)
True clusters



EM does well



kmeans totally misses