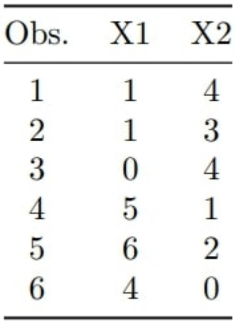
Q42

Rahul Atre

2023-11-28

Q42 [Clustering]

In this problem, you will perform K-means clustering manually, with K = 2, on a small example with n = 6 observations and p = 2 features. The observations are as follows.

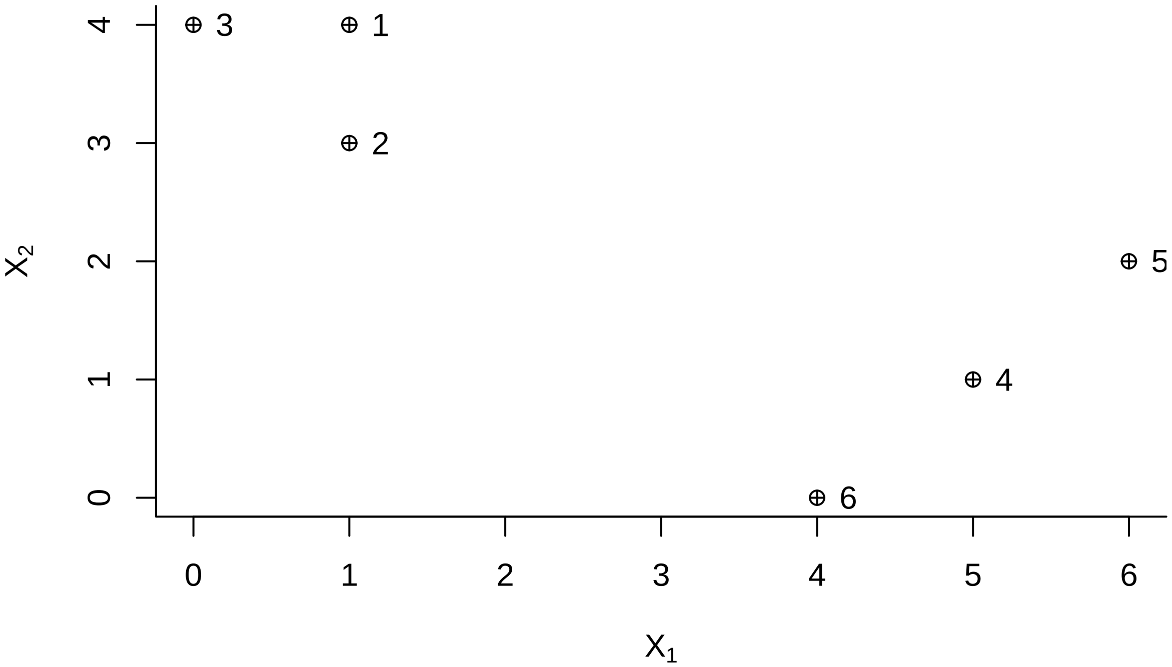


Obs.	X1	X2
1	1	4
2	1	3
3	0	4
4	5	1
5	6	2
6	4	0

1. Plot the observations.

Let us first store the observation and values in a dataset:

```
obs_xValues = cbind(c(1, 1, 0, 5, 6, 4), c(4, 3, 4, 1, 2, 0))
plot(obs_xValues[,1], obs_xValues[,2], xlab = expression(X[1]), ylab = expression(X[2]), pch = 10, bty =
text(obs_xValues[, 1] + 0.20, obs_xValues[, 2], 1:6)
```

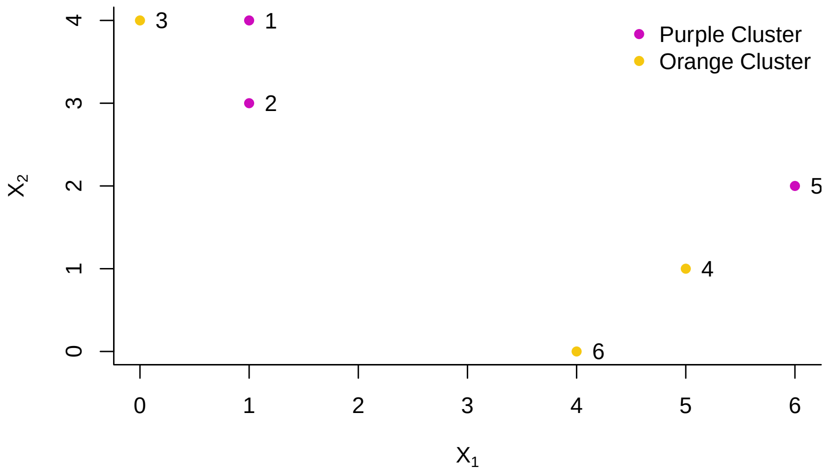


2. Randomly assign a cluster label to each observation. You can use the sample() command in R to do this. Report the cluster labels for each observation.

```
set.seed(15)

cluster_label = sample(2, nrow(obs_xValues), replace = TRUE)
obs_xValues = cbind(obs_xValues, cluster_label)

plot(obs_xValues[,1], obs_xValues[,2], xlab = expression(X[1]), ylab = expression(X[2]), col = (cluster_
text(obs_xValues[,1]+0.20, obs_xValues[,2], 1:6)
legend("topright", legend = c("Purple Cluster", "Orange Cluster"), col = c(1,2) + 5, bty = "n", pch = 1
```



```
# Purple -> 1, Orange -> 2
```

3. Compute the centroid for each cluster.

```
purple_centroid = colMeans(obs_xValues[obs_xValues[,3] == 1,, drop = FALSE])[1:2]
purple_centroid
```

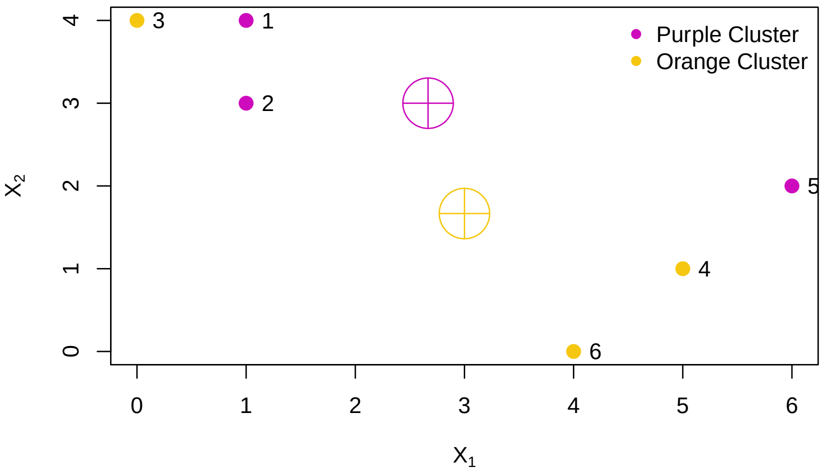
```
##
## 2.666667 3.000000
```

```
orange_centroid = colMeans(obs_xValues[obs_xValues[,3] == 2,, drop = FALSE])[1:2]
orange_centroid
```

```
##
## 3.000000 1.666667
```

From the above function call, we can see that the purple cluster's centroid is (2.67, 3), and the orange cluster's centroid is (3, 1.67). Plotting this in our cluster graph we get:

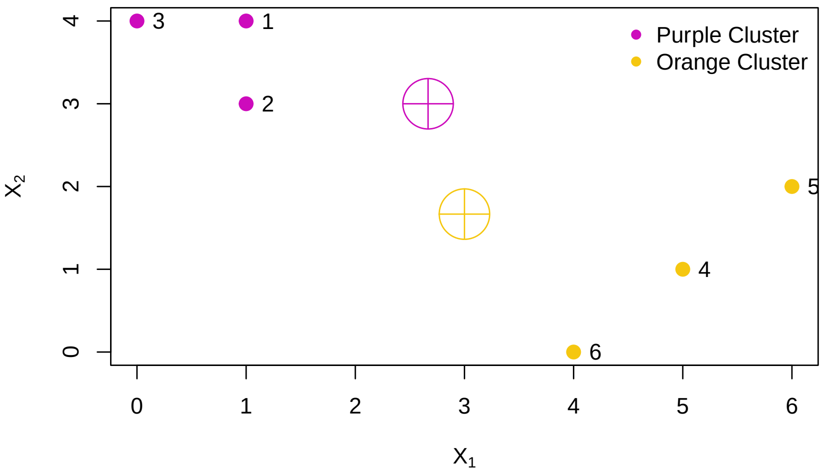
```
plot(obs_xValues[,1], obs_xValues[,2], col=(cluster_label + 5), pch = 20, cex = 2, xlab = expression(X[
legend("topright", legend = c("Purple Cluster", "Orange Cluster"), col = c(1,2) + 5, bty = "n", pch = 1
text(obs_xValues[,1]+0.20, obs_xValues[,2], 1:6)
points(purple_centroid[1], purple_centroid[2], col = 6, pch = 10, cex = 5)
points(orange_centroid[1], orange_centroid[2], col = 7, pch = 10, cex = 5)
```



4. Assign each observation to the centroid to which it is closest, in terms of Euclidean distance. Report the cluster labels for each observation.

```r
assigned_labels = c(1, 1, 1, 2, 2, 2)
obs_xValues[, 3] = assigned_labels

plot(obs_xValues[,1], obs_xValues[,2], col=(assigned_labels + 5), pch = 20, cex = 2, xlab = expression(
legend("topright", legend = c("Purple Cluster", "Orange Cluster"), col = c(1,2) + 5, bty = "n", pch = 1
text(obs_xValues[,1]+0.20, obs_xValues[,2], 1:6)
points(purple_centroid[1], purple_centroid[2], col = 6, pch = 10, cex = 5)
points(orange_centroid[1], orange_centroid[2], col = 7, pch = 10, cex = 5)
```



5. Repeat 3. and 4. until the answers obtained stop changing.

```
purple_centroid = colMeans(obs_xValues[obs_xValues[,3] == 1,, drop = FALSE])[1:2]
purple_centroid
```

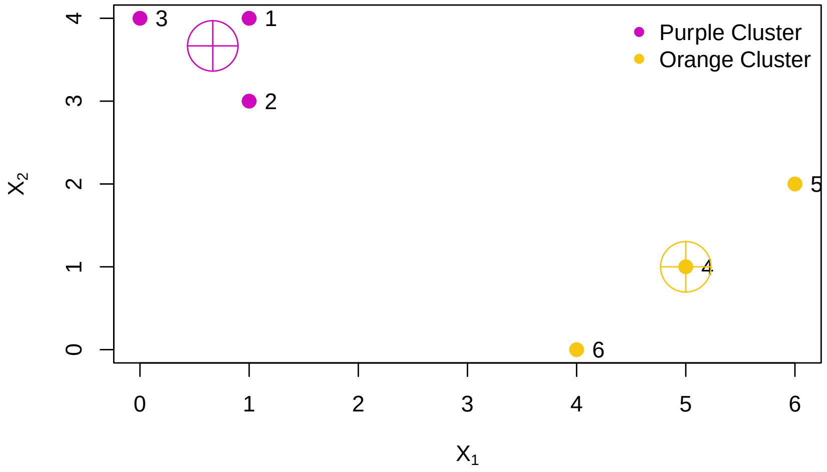
```
##
## 0.6666667 3.6666667
```

```
orange_centroid = colMeans(obs_xValues[obs_xValues[,3] == 2,, drop = FALSE])[1:2]
orange_centroid
```

```
##
## 5 1
```

```
assigned_labels = c(1, 1, 1, 2, 2, 2)
obs_xValues[, 3] = assigned_labels

plot(obs_xValues[,1], obs_xValues[,2], col=(assigned_labels + 5), pch = 20, cex = 2, xlab = expression(
legend("topright", legend = c("Purple Cluster", "Orange Cluster"), col = c(1,2) + 5, bty = "n", pch = 1
text(obs_xValues[,1]+0.20, obs_xValues[,2], 1:6)
points(purple_centroid[1], purple_centroid[2], col = 6, pch = 10, cex = 5)
points(orange_centroid[1], orange_centroid[2], col = 7, pch = 10, cex = 5)
```

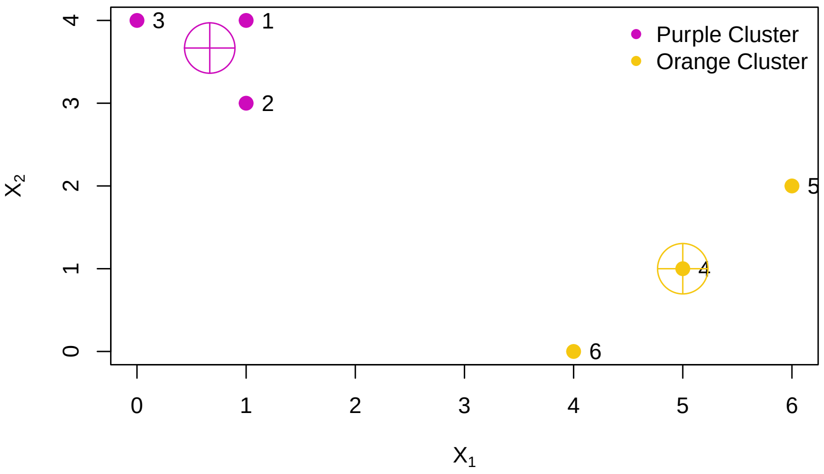


As we can see, the centroid value gets closer to the observations with an increase in the number of iterations.

6. In your plot from 1., color the observations according to the cluster labels obtained.

Same plot as (5):

```
plot(obs_xValues[,1], obs_xValues[,2], col=(assigned_labels + 5), pch = 20, cex = 2, xlab = expression(
legend("topright", legend = c("Purple Cluster", "Orange Cluster"), col = c(1,2) + 5, bty = "n", pch = 1
text(obs_xValues[,1]+0.20, obs_xValues[,2], 1:6)
points(purple_centroid[1], purple_centroid[2], col = 6, pch = 10, cex = 5)
points(orange_centroid[1], orange_centroid[2], col = 7, pch = 10, cex = 5)
```


3
1
2
5
4
6
Purple Cluster
Orange Cluster
X_2
X_1
0
1
2
3
4
5
6