Bruce Campbell ST-617 Homework 5

Wed Jul 27 20:21:25 2016

```
rm(list = ls())
set.seed(7)
```

Chapter 10

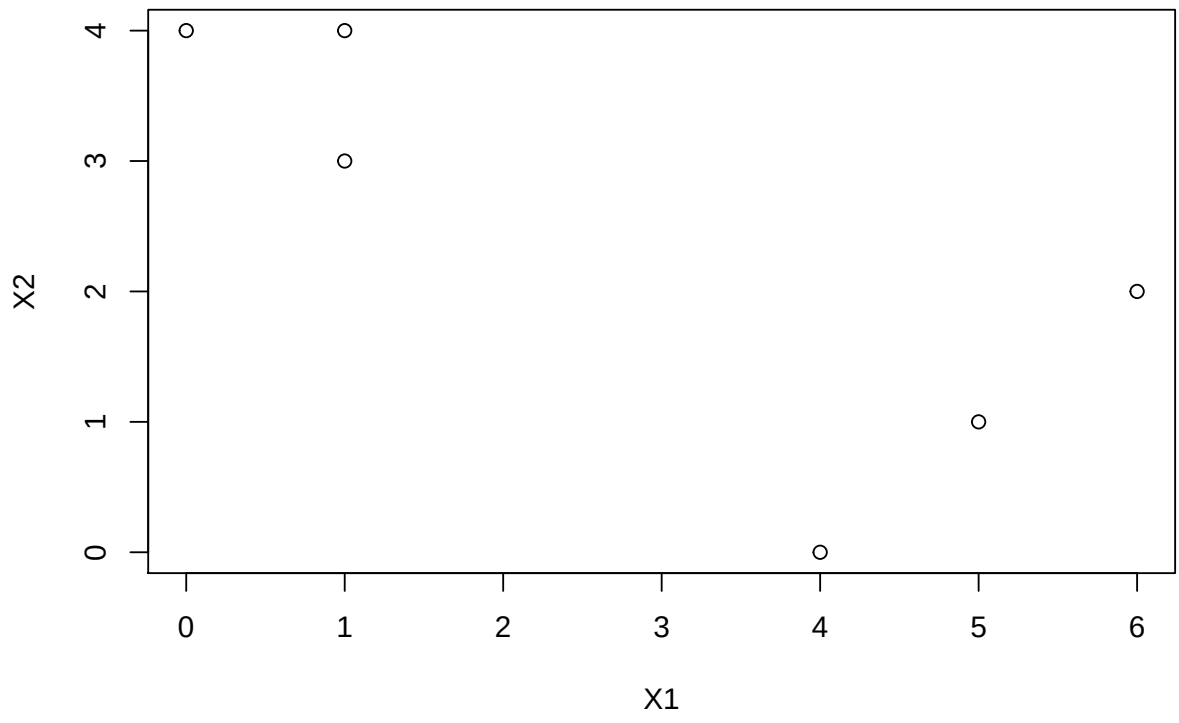
Problem 3

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.

a)

Plot the observations.

```
X1 <- c(1, 1, 0, 5, 6, 4)
X2 <- c(4, 3, 4, 1, 2, 0)
plot(X1, X2)
```



b)

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.

```
DF <- data.frame(X1 = X1, X2 = X2)
class1 <- sample(nrow(DF), floor(nrow(DF)/2))

class <- matrix(0, nrow = nrow(DF), ncol = 1)
class[class1] <- 1
DF$class <- class
library(pander)
pander(DF)
```

X1	X2	class
1	4	1
1	3	1
0	4	0
5	1	0
6	2	0
4	0	1

c)

Compute the centroid for each cluster.

```
DFClass1 <- DF[DF$class == 1, ]
DFClass1$class <- NULL
centroid1 <- colMeans(as.matrix(DFClass1))


DFClass0 <- DF[DF$class == 0, ]
DFClass0$class <- NULL
centroid0 <- colMeans(as.matrix(DFClass0))
```

d)

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

```
DFDist <- rbind(DFClass0, DFClass1, centroid0, centroid1)

# Boo Duplicate Row names not allowed. Oh well we understand why
index <- (c(rep("class 0", nrow(DFClass0)), c(rep("class 1", nrow(DFClass1)),
    "Centroid0", "Centroid1")))
# row.names(DFDist)<-index

row.names(DFDist) <- c("C01", "C02", "C03", "C11", "C12", "C13", "Centroid0",
    "Centroid1")
```

```
dmat <- as.matrix(dist(DFDist))

centroid0Index <- which(index == "Centroid0")

centroid1Index <- which(index == "Centroid1")

indicatorCentroid1_isCloser <- dmat[centroid1Index, ] < dmat[centroid0Index,
    ]
indicatorCentroid1_isCloser
```

```
##       C01       C02       C03       C11       C12       C13 Centroid0
##      TRUE     FALSE     FALSE      TRUE      TRUE     FALSE     FALSE
## Centroid1
##      TRUE
```

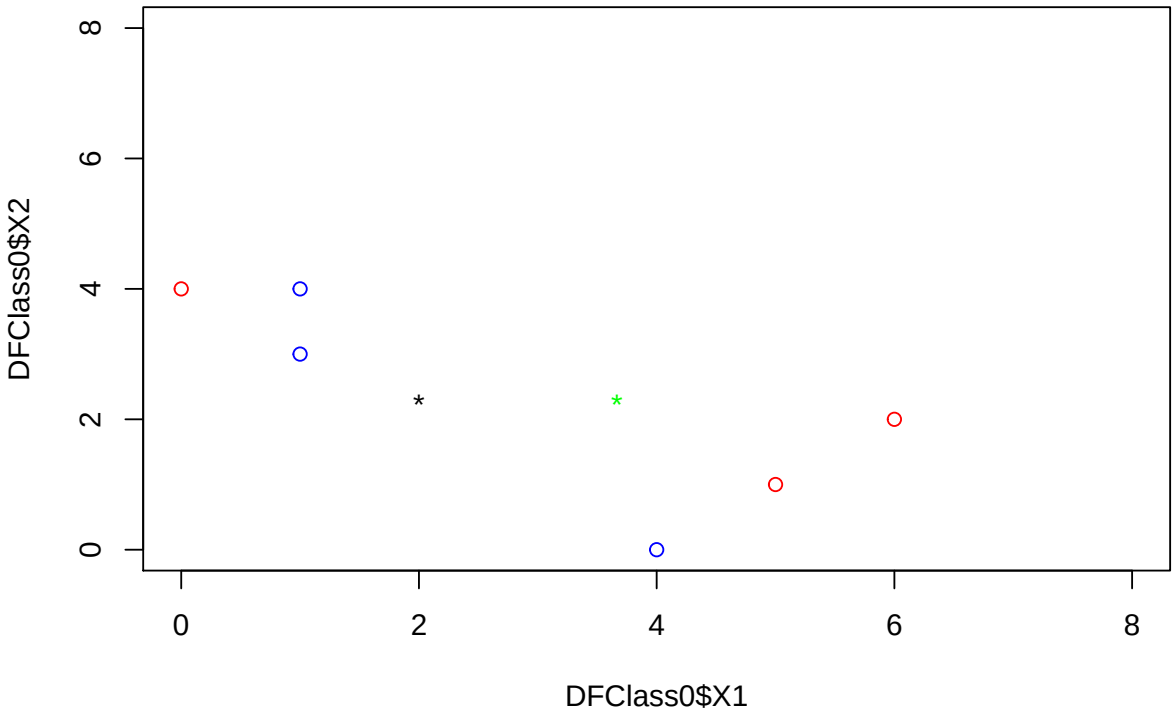
```
cluster1 <- which(indicatorCentroid1_isCloser[c(-7, -8)])
cluster0 <- which(indicatorCentroid1_isCloser[c(-7, -8)] == FALSE)

pander(cluster0, caption = "Cluster 0")
```

C02	C03	C13
2	3	6

```
pander(cluster1, caption = "Cluster 1")
```

C01	C11	C12
1	4	5

```
plot(DFClass0$X1, DFClass0$X2, col = "red", xlim = c(0, 8), ylim = c(0, 8))
points(DFClass1$X1, DFClass1$X2, col = "blue")
points(centroid0[1], centroid0[2], pch = "*", col = "green")
points(centroid1[1], centroid1[2], pch = "*", col = "black")
```

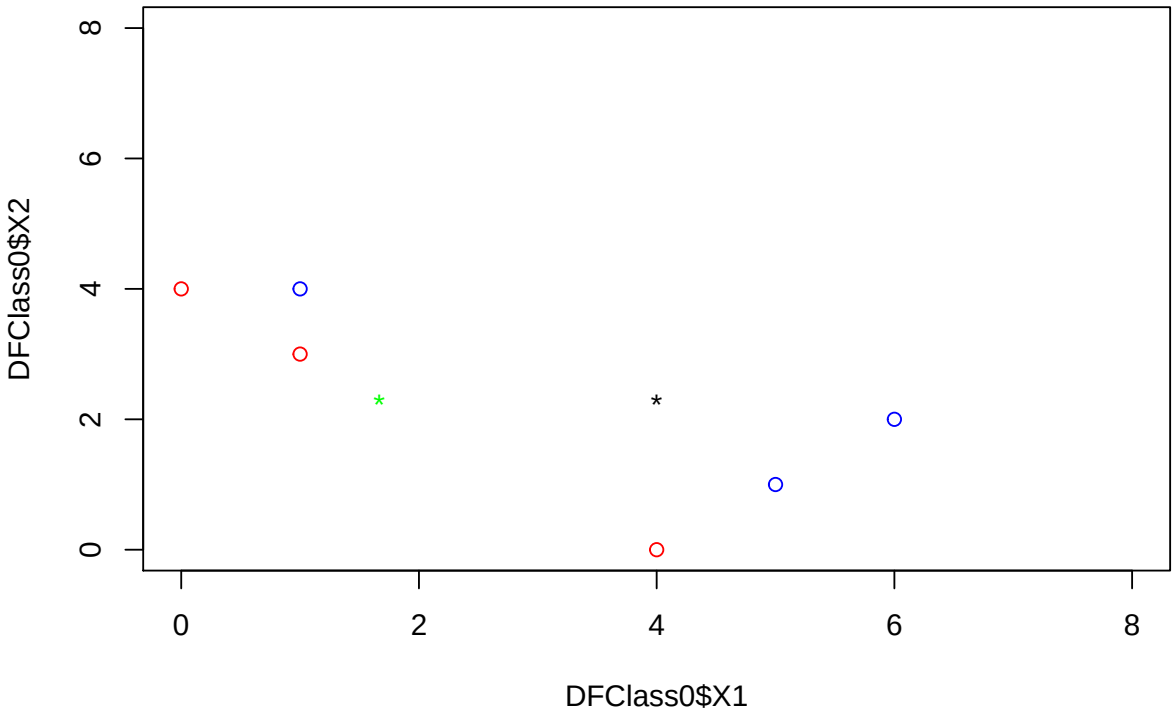


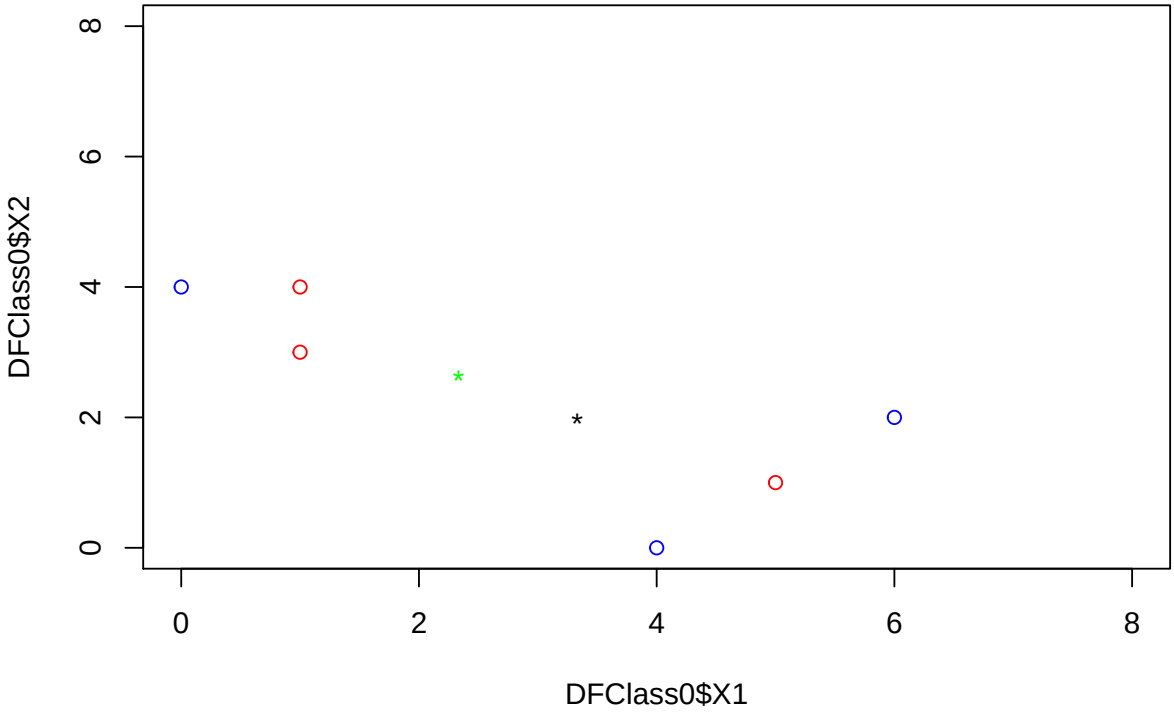
e)

Repeat (c) and (d) until the answers obtained stop changing.

```
oldCluster0 <- cluster0
oldCluster1 <- cluster1
numIter = 10
for (i in 1:numIter) {
    DFClass1 <- DF[cluster1, ]
    centroid1 <- colMeans(as.matrix(DFClass1))

    DFClass0 <- DF[cluster0, ]
    centroid0 <- colMeans(as.matrix(DFClass0))
    DFDist <- rbind(DFClass0, DFClass1, centroid0, centroid1)

    index <- (c(rep("class 0", nrow(DFClass0)), c(rep("class 1", nrow(DFClass1)),
        "Centroid0", "Centroid1")))

    row.names(DFDist) <- c("C01", "C02", "C03", "C11", "C12", "C13", "Centroid0",
        "Centroid1")

    dmat <- as.matrix(dist(DFDist))

    centroid0Index <- which(index == "Centroid0")
```

```
    centroid1Index <- which(index == "Centroid1")

    indicatorCentroid1_isCloser <- dmat[centroid1Index, ] < dmat[centroid0Index,
        ]

    cluster1 <- which(indicatorCentroid1_isCloser[c(-7, -8)])
    cluster0 <- which(indicatorCentroid1_isCloser[c(-7, -8)] == FALSE)

    pander(cluster0, caption = "Cluster 0")

    pander(cluster1, caption = "Cluster 1")

    numIter <- numIter - 1
    plot(DFClass0$X1, DFClass0$X2, col = "red", xlim = c(0, 8), ylim = c(0,
        8))
    points(DFClass1$X1, DFClass1$X2, col = "blue")
    points(centroid0[1], centroid0[2], pch = "*", col = "green")
    points(centroid1[1], centroid1[2], pch = "*", col = "black")

    numIter
}
```


DFClass0$X2
0
2
4
6
8
DFClass0$X1
0
2
4
6
8

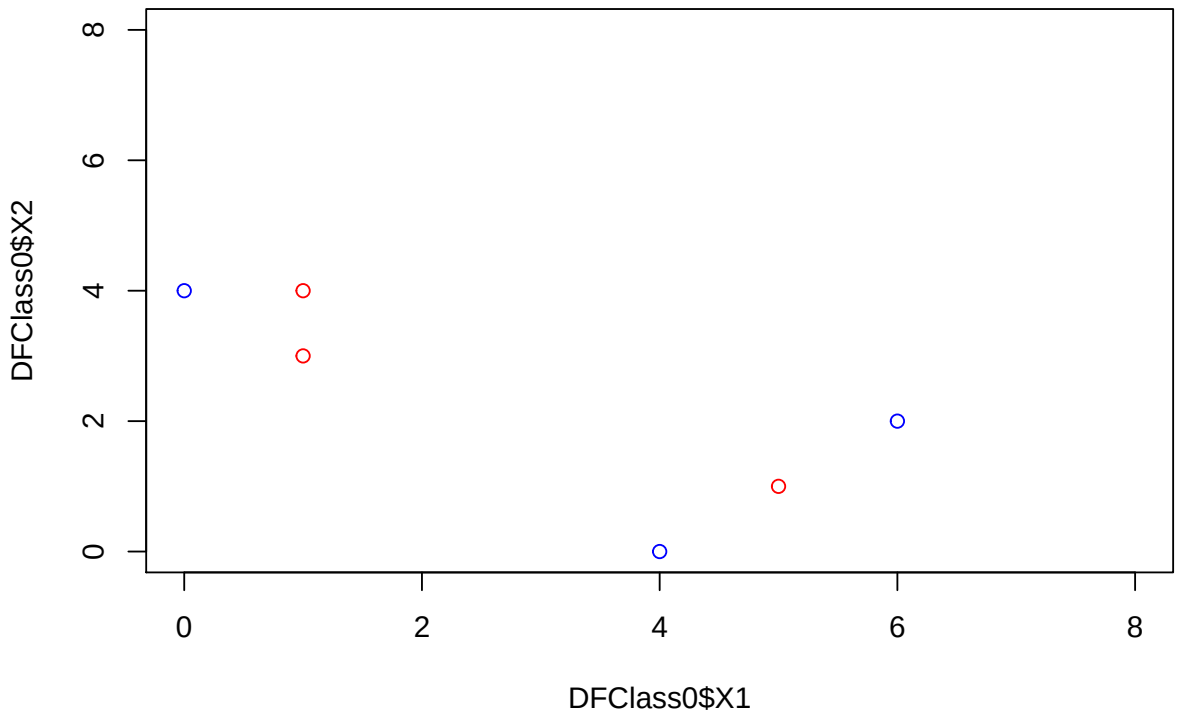

DFClass0$X2
0
2
4
6
8
DFClass0$X1
0
2
4
6
8

```
# We don't have to put the code above in a loop out of luck.  We'd like to
# revisit this and put the code in a loop.
```

f)

In your plot from (a), color the observations according to the cluster labels obtained.

```
plot(DFClass0$X1, DFClass0$X2, col = "red", xlim = c(0, 8), ylim = c(0, 8))
points(DFClass1$X1, DFClass1$X2, col = "blue")
```



We have a bug or an unlucky situation