DA6

2025-06-25

10.5 Lab 2: Clustering

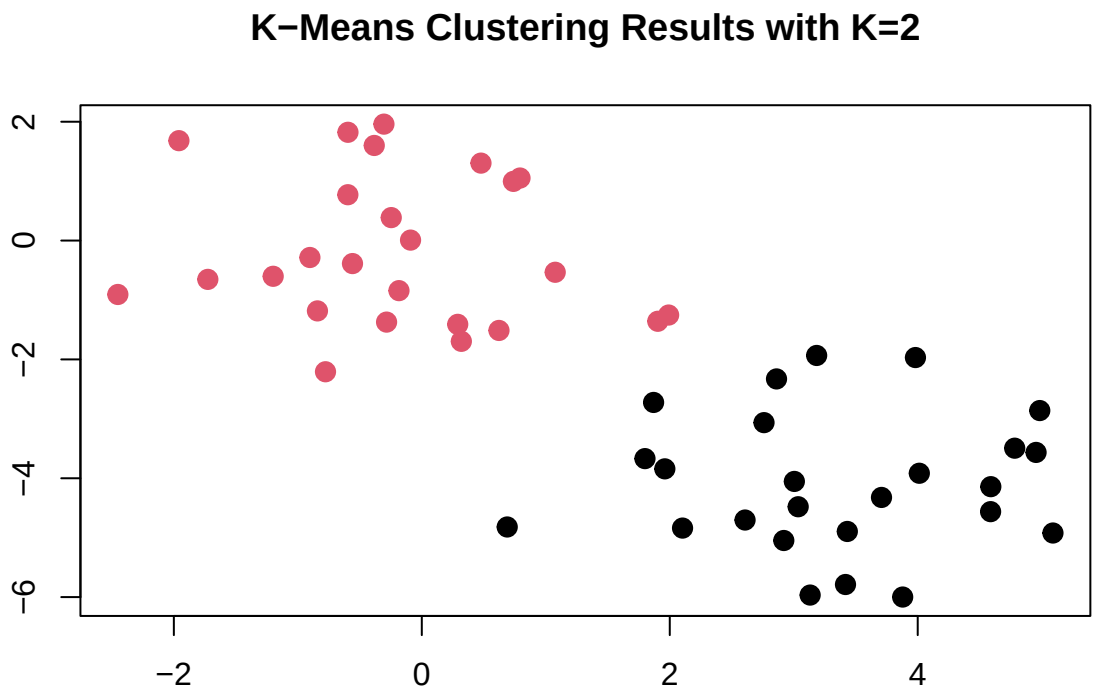
```
set.seed(2)
x = matrix(rnorm(50*2), ncol = 2)
x[1:25,1]=x[1:25,1]+3
x[1:25,2]=x[1:25,2]-4

x
```

```
##              [,1]         [,2]
##  [1,]  2.10308545 -4.838287148
##  [2,]  3.18484918 -1.933698644
##  [3,]  4.58784533 -4.562247053
##  [4,]  1.86962433 -2.724284488
##  [5,]  2.91974824 -5.047572627
##  [6,]  3.13242028 -5.965878241
##  [7,]  3.70795473 -4.322971094
##  [8,]  2.76030198 -3.064137473
##  [9,]  4.98447394 -2.860770197
## [10,]  2.86121299 -2.328381233
## [11,]  3.41765075 -5.788242207
## [12,]  3.98175278 -1.968757481
## [13,]  2.60730464 -4.703144333
## [14,]  1.96033102 -3.841835237
## [15,]  4.78222896 -3.493765203
## [16,]  0.68893092 -4.819995106
## [17,]  3.87860458 -5.998846995
## [18,]  3.03580672 -4.479292591
## [19,]  4.01282869 -3.915820096
## [20,]  3.43226515 -4.895486611
## [21,]  5.09081921 -4.921275666
## [22,]  1.80007418 -3.669550497
## [23,]  4.58963820 -4.141660809
## [24,]  4.95465164 -3.565152238
## [25,]  3.00493778 -4.053722626
## [26,] -2.45170639 -0.907110376
## [27,]  0.47723730  1.303512232
## [28,] -0.59655817  0.771789776
## [29,]  0.79220327  1.052525595
## [30,]  0.28963671 -1.410038341
## [31,]  0.73893860  0.995984590
## [32,]  0.31896040 -1.695764903
## [33,]  1.07616435 -0.533372143
## [34,] -0.28415772 -1.372269451
```

```
## [35,] -0.77667527 -2.207919779
## [36,] -0.59566050  1.822122519
## [37,] -1.72597978 -0.653393411
## [38,] -0.90258448 -0.284681219
## [39,] -0.55906191 -0.386949604
## [40,] -0.24651257  0.386694975
## [41,] -0.38358623  1.600390852
## [42,] -1.95910318  1.681154956
## [43,] -0.84170506 -1.183606388
## [44,]  1.90354747 -1.358457254
## [45,]  0.62249393 -1.512670795
## [46,]  1.99092044 -1.253104899
## [47,] -0.30548372  1.959357077
## [48,] -0.09084424  0.007645872
## [49,] -0.18416145 -0.842615198
## [50,] -1.19876777 -0.601160105
```

```
km.out = kmeans(x, 2, nstart =20)
plot(x, col=(km.out$cluster), main="K-Means Clustering Results with K=2", xlab="", ylab="", pch=20, cex=
```




```
set.seed(4)
km.out=kmeans(x,3, nstart =20)
km.out
```

```
## K-means clustering with 3 clusters of sizes 10, 23, 17
```

```
##
## Cluster means:
##         [,1]        [,2]
## 1  2.3001545 -2.69622023
## 2 -0.3820397 -0.08740753
## 3  3.7789567 -4.56200798
##
## Clustering vector:
##  [1] 3 1 3 1 3 3 3 1 3 1 3 1 3 1 3 1 3 3 3 3 3 1 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2
## [39] 2 2 2 2 2 1 2 1 2 2 2 2
##
## Within cluster sum of squares by cluster:
## [1] 19.56137 52.67700 25.74089
##  (between_SS / total_SS =  79.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

nstart = 20 means run the k-means algorithm 20 times, each time with a new random set of starting centers, and then keep the solution that has the lowest total within-cluster sum of squares.

km.out$cluster: a length-n vector assigning each row of x to cluster 1, 2, or 3.

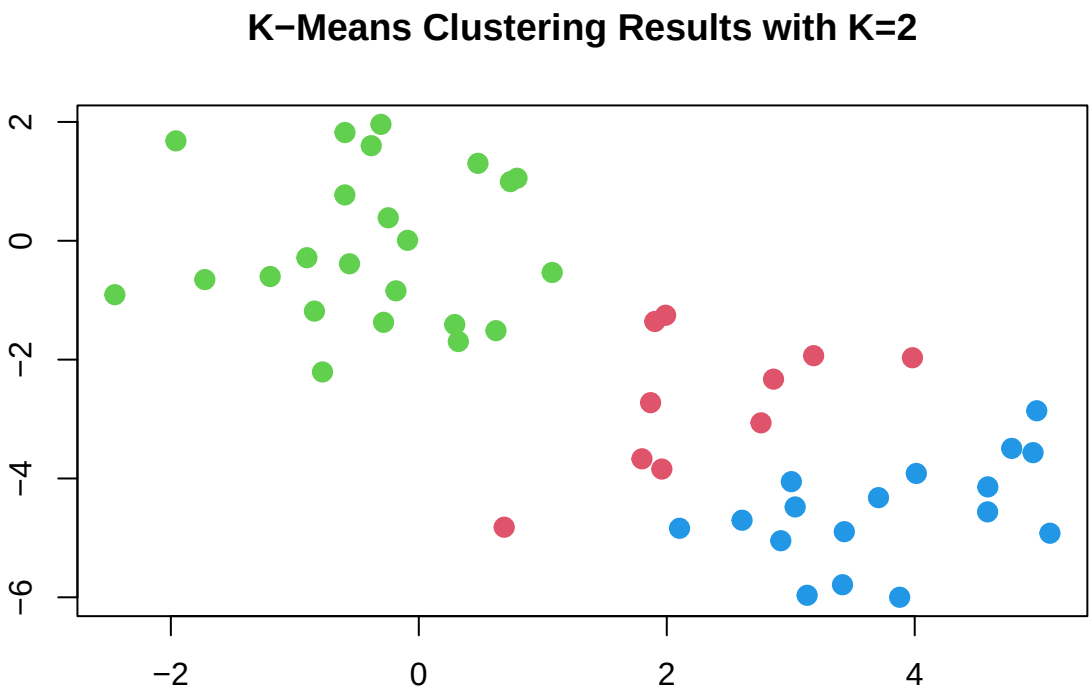
km.out$centers: the coordinates of the three cluster centroids in feature space.

km.out$withinss: vector of within-cluster sum of squares for each cluster.

km.out$tot.withinss: the total within-cluster sum of squares (the quantity that k-means tries to minimize).

km.out$betweenss: the between-cluster sum of squares.

```
plot(x, col=(km.out$cluster+1), main="K-Means Clustering Results with K=2", xlab="", ylab="", pch=20, c
```



it is better to use a large nstart (20/50), so it would be a real good graph