**> sort(abs(r))[seq(1,64,by=2)]**

**1. 0.00957748 0.01439009 0.03435535 0.04091691 0.06090397 0.08472743**

**2. 0.29305124 0.30104166**

**3. 0.36447305 0.37251617 0.37525826**

**4. 0.44448296**

**5. 0.72440126 0.77557249**

**6. 0.88574747 0.89713554 0.90784760 0.91727340 0.92704371 0.93576442 0.94234741 0.95100221 0.95773956 0.99684171 0.99711810 0.99871829 0.99963811 0.99968038**

**7. 1.00000000 1.00000000 1.00000000 1.00000000**

**> matrix(c,ncol=2,byrow=T)**

**[,1] [,2]**

**[1,] 4 1**

**[2,] 3 1**

**[3,] 8 6**

**[4,] 6 1**

**Cluster 1: 5, 7**

**Cluster 2: 1, 8**

**Cluster 3: 2**

**Cluster 4: 3, 4, 6**

**[5,] 8 3**

**[6,] 8 4**

**[7,] 2 1**

**[8,] 8 5**

**[9,] 8 2**

**[10,] 5 1**

**[11,] 8 7**

**[12,] 7 1**

**[13,] 7 2**

**[14,] 5 2**

**[15,] 7 4**

**[16,] 7 3**

**[17,] 7 6**

**[18,] 5 4**

**[19,] 5 3**

**[20,] 6 5**

**[21,] 6 2**

**[22,] 3 2**

**[23,] 4 2**

**[24,] 7 5**

**[25,] 8 1**

**[26,] 6 4**

**[27,] 6 3**

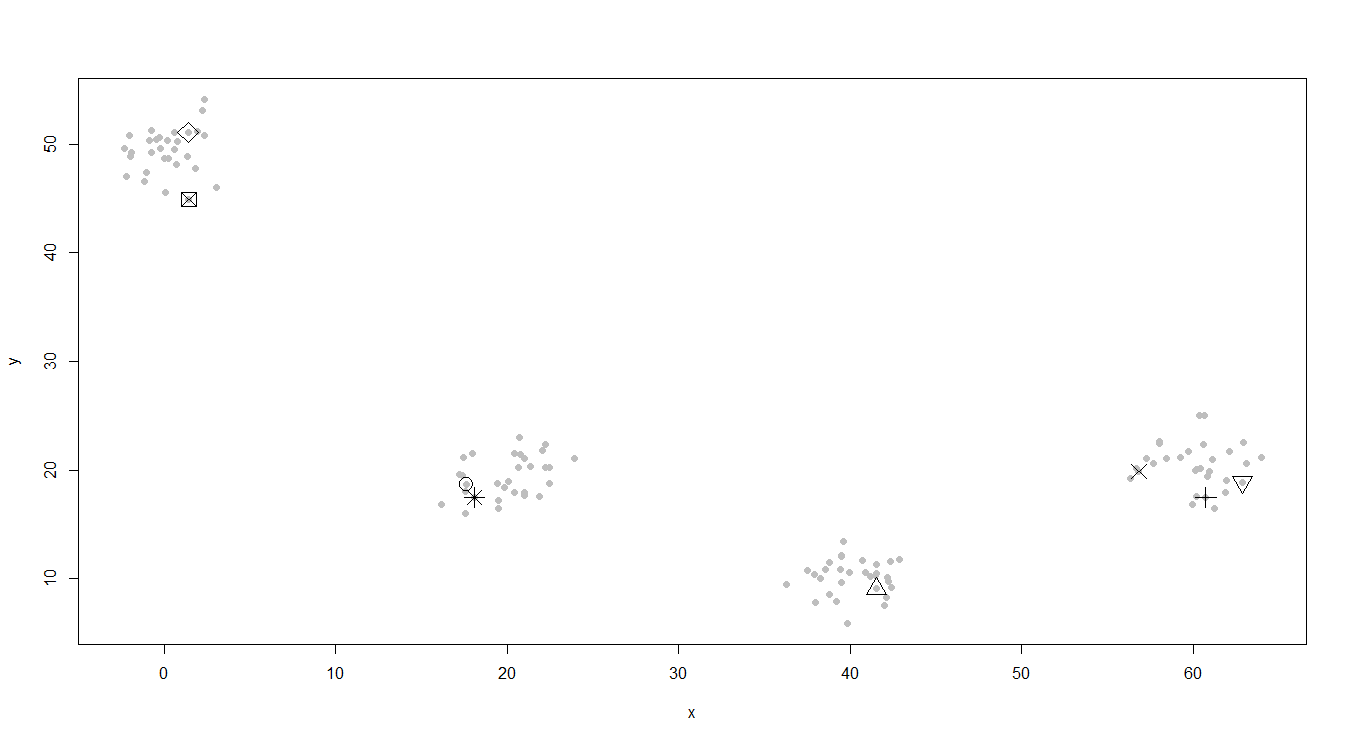
**[28,] 4 3**

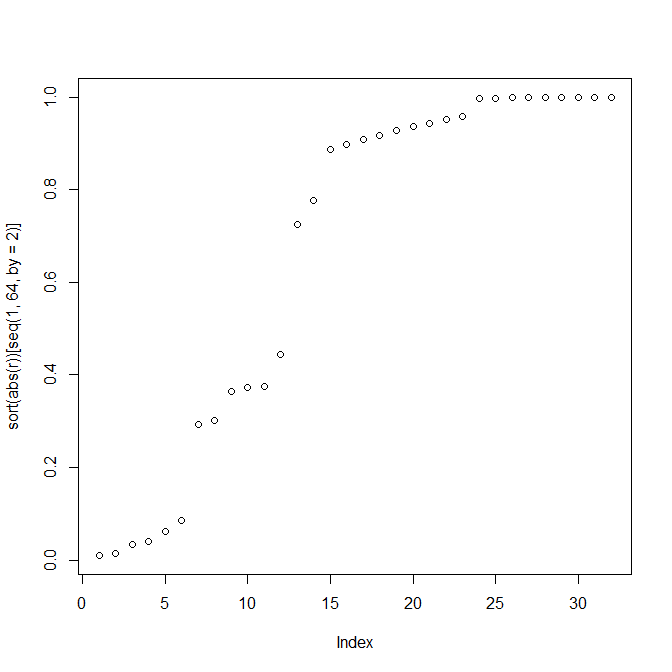
**[29,] 1 1**

**[30,] 1 1**

**[31,] 1 1**

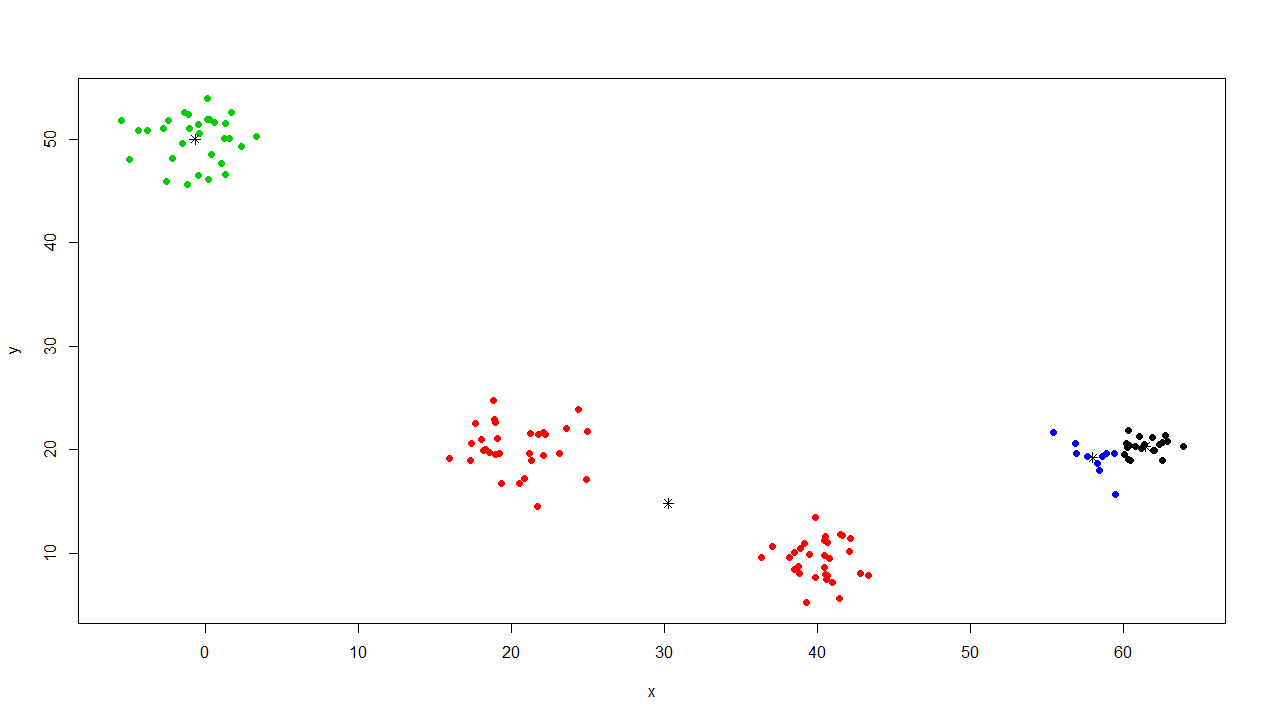
**[32,] 1 1**



****

Algorithm:

1. **First we select a larger number (n) of centres than required (k) to ensure that there are at least one centroid in each cluster.**
2. **Then we calculate the distance of all points from these n centres and form our distance matrix of dimensions nrow(x) \* n.**
3. **Then we form the correlation matrix between the columns of distance matrix.**
4. **Then we select the k centres such that the correlation between their distance vectors is least.**



* Ordinary k-means attains a local minima : But this method gives proper starting centres