ix. Exercise 10.7.10

Kaarthik Sundaramoorthy, Sahil Shah and Vidhi Shah

7/25/2020

In this problem, you will generate simulated data, and then perform PCA and K-means clustering on the data.

(a) Generate a simulated data set with 20 observations in each of three classes (i.e. 60 observations total), and 50 variables.

Hint: There are a number of functions in `R` that you can use to generate data. One example is the `rnorm()` function; `runif()` is another option. Be sure to add a mean shift to the observations in each class so that there are three distinct classes.

```
set.seed(3)
x = matrix(rnorm(20*3*50, mean = 0, sd = 0.01), ncol = 50)
x[1:20, 2] = 1
x[21:40, 1] = 2
x[21:40, 2] = 2
x[41:60, 1] = 1
true.labels <- c(rep(1,20), rep(2,20), rep(3,20))
summary(x)
```

```
##        V1                  V2                  V3                 
##  Min.   :-0.012189   Min.   :-0.018977   Min.   :-0.0240366  
##  1st Qu.: 0.002394   1st Qu.: 0.006067   1st Qu.:-0.0098610  
##  Median : 1.000000   Median : 1.000000   Median : 0.0018459  
##  Mean   : 0.999443   Mean   : 1.000210   Mean   : 0.0003014  
##  3rd Qu.: 2.000000   3rd Qu.: 2.000000   3rd Qu.: 0.0091260  
##  Max.   : 2.000000   Max.   : 2.000000   Max.   : 0.0263505  
##        V4                   V5                  V6                 
##  Min.   :-0.0168837   Min.   :-0.022347   Min.   :-0.0239845  
##  1st Qu.:-0.0089555   1st Qu.:-0.004890   1st Qu.:-0.0052198  
##  Median :-0.0027811   Median : 0.002592   Median : 0.0013762  
##  Mean   :-0.0003856   Mean   : 0.002060   Mean   : 0.0005896  
##  3rd Qu.: 0.0065298   3rd Qu.: 0.008387   3rd Qu.: 0.0052381  
##  Max.   : 0.0267663   Max.   : 0.025362   Max.   : 0.0351930  
##        V7                   V8                   V9                 
##  Min.   :-0.0204848   Min.   :-0.0266570   Min.   :-0.025459  
##  1st Qu.:-0.0064525   1st Qu.:-0.0071612   1st Qu.:-0.006029  
##  Median :-0.0025617   Median : 0.0004302   Median :-0.000946  
##  Mean   : 0.0003492   Mean   : 0.0008622   Mean   :-0.001119  
##  3rd Qu.: 0.0060304   3rd Qu.: 0.0073419   3rd Qu.: 0.002838  
##  Max.   : 0.0216384   Max.   : 0.0320059   Max.   : 0.019609  
##       V10                 V11                 V12
```

```
##  Min.   :-0.021113   Min.   :-0.023779   Min.   :-0.0221498
##  1st Qu.:-0.008293   1st Qu.:-0.007162   1st Qu.:-0.0066534
##  Median :-0.001731   Median :-0.001435   Median :-0.0006834
##  Mean   :-0.001233   Mean   :-0.001506   Mean   :-0.0014067
##  3rd Qu.: 0.004915   3rd Qu.: 0.005217   3rd Qu.: 0.0052413
##  Max.   : 0.017776   Max.   : 0.016644   Max.   : 0.0141310
##       V13                 V14                 V15
##  Min.   :-0.0305330   Min.   :-0.022668   Min.   :-0.0239161
##  1st Qu.:-0.0072571   1st Qu.:-0.007521   1st Qu.:-0.0059992
##  Median :-0.0003743   Median : 0.002084   Median : 0.0013212
##  Mean   :-0.0005633   Mean   : 0.001228   Mean   : 0.0008891
##  3rd Qu.: 0.0061810   3rd Qu.: 0.007823   3rd Qu.: 0.0072997
##  Max.   : 0.0178642   Max.   : 0.021897   Max.   : 0.0175596
##       V16                 V17                 V18
##  Min.   :-0.0155768   Min.   :-0.030563   Min.   :-0.0324049
##  1st Qu.:-0.0071337   1st Qu.:-0.009377   1st Qu.:-0.0051215
##  Median : 0.0007607   Median : 0.000117   Median : 0.0007584
##  Mean   : 0.0006789   Mean   :-0.001013   Mean   : 0.0008674
##  3rd Qu.: 0.0077086   3rd Qu.: 0.006890   3rd Qu.: 0.0085533
##  Max.   : 0.0222890   Max.   : 0.025955   Max.   : 0.0216652
##       V19                 V20                 V21
##  Min.   :-0.023509   Min.   :-0.022410   Min.   :-1.971e-02
##  1st Qu.:-0.008053   1st Qu.:-0.009815   1st Qu.:-3.920e-03
##  Median :-0.001083   Median :-0.002111   Median : 8.639e-05
##  Mean   :-0.001280   Mean   :-0.002454   Mean   : 7.599e-04
##  3rd Qu.: 0.004840   3rd Qu.: 0.003942   3rd Qu.: 5.594e-03
##  Max.   : 0.025236   Max.   : 0.016868   Max.   : 2.138e-02
##       V22                 V23                  V24
##  Min.   :-0.0205388   Min.   :-0.0226143   Min.   :-0.0208815
##  1st Qu.:-0.0081826   1st Qu.:-0.0054177   1st Qu.:-0.0055456
##  Median :-0.0004464   Median : 0.0008800   Median : 0.0005435
##  Mean   :-0.0016794   Mean   : 0.0002786   Mean   : 0.0011279
##  3rd Qu.: 0.0040335   3rd Qu.: 0.0068651   3rd Qu.: 0.0095111
##  Max.   : 0.0169311   Max.   : 0.0205141   Max.   : 0.0210395
##       V25                 V26                 V27
##  Min.   :-0.0284827   Min.   :-0.016281   Min.   :-0.0186121
##  1st Qu.:-0.0058732   1st Qu.:-0.006555   1st Qu.:-0.0076783
##  Median : 0.0006802   Median : 0.002157   Median : 0.0001106
##  Mean   : 0.0005239   Mean   : 0.001012   Mean   :-0.0001298
##  3rd Qu.: 0.0070868   3rd Qu.: 0.007496   3rd Qu.: 0.0070290
##  Max.   : 0.0308400   Max.   : 0.023684   Max.   : 0.0188451
##       V28                 V29                 V30
##  Min.   :-0.024444   Min.   :-0.0231533   Min.   :-0.0292324
##  1st Qu.:-0.009237   1st Qu.:-0.0079144   1st Qu.:-0.0058328
##  Median :-0.002114   Median :-0.0003922   Median : 0.0005926
##  Mean   :-0.002133   Mean   :-0.0003543   Mean   :-0.0005039
##  3rd Qu.: 0.004640   3rd Qu.: 0.0073276   3rd Qu.: 0.0081039
##  Max.   : 0.019240   Max.   : 0.0220601   Max.   : 0.0175659
##       V31                 V32                  V33
##  Min.   :-0.0224454   Min.   :-0.0244356   Min.   :-0.0213458
##  1st Qu.:-0.0070196   1st Qu.:-0.0067233   1st Qu.:-0.0053071
##  Median : 0.0004843   Median :-0.0020708   Median :-0.0001234
##  Mean   : 0.0008579   Mean   :-0.0008185   Mean   : 0.0006103
##  3rd Qu.: 0.0074904   3rd Qu.: 0.0064132   3rd Qu.: 0.0055437
```

```
##  Max.   : 0.0238513   Max.   : 0.0229652   Max.   : 0.0245354
##       V34                  V35                  V36
##  Min.   :-0.0215554   Min.   :-0.0231652   Min.   :-0.020914
##  1st Qu.:-0.0070747   1st Qu.:-0.0052465   1st Qu.:-0.009241
##  Median : 0.0010765   Median : 0.0001697   Median :-0.001353
##  Mean   : 0.0009507   Mean   : 0.0002721   Mean   :-0.002002
##  3rd Qu.: 0.0074772   3rd Qu.: 0.0051915   3rd Qu.: 0.004266
##  Max.   : 0.0300327   Max.   : 0.0345855   Max.   : 0.015708
##       V37                  V38                  V39
##  Min.   :-0.0178034   Min.   :-0.0237843   Min.   :-0.0183610
##  1st Qu.:-0.0047485   1st Qu.:-0.0072070   1st Qu.:-0.0075844
##  Median : 0.0004186   Median :-0.0008254   Median : 0.0005050
##  Mean   : 0.0004455   Mean   :-0.0003707   Mean   : 0.0005592
##  3rd Qu.: 0.0055357   3rd Qu.: 0.0076262   3rd Qu.: 0.0074734
##  Max.   : 0.0250246   Max.   : 0.0206269   Max.   : 0.0234559
##       V40                  V41                  V42
##  Min.   :-0.0270240   Min.   :-2.037e-02   Min.   :-0.0275057
##  1st Qu.:-0.0058642   1st Qu.:-6.569e-03   1st Qu.:-0.0072311
##  Median : 0.0003879   Median :-4.152e-05   Median :-0.0006648
##  Mean   : 0.0009678   Mean   : 3.470e-04   Mean   :-0.0012655
##  3rd Qu.: 0.0059197   3rd Qu.: 5.181e-03   3rd Qu.: 0.0043277
##  Max.   : 0.0250008   Max.   : 2.593e-02   Max.   : 0.0204635
##       V43                  V44                  V45
##  Min.   :-0.0246905   Min.   :-0.0203512   Min.   :-0.021569
##  1st Qu.:-0.0068120   1st Qu.:-0.0082840   1st Qu.:-0.006307
##  Median : 0.0003802   Median : 0.0011675   Median : 0.001055
##  Mean   :-0.0005864   Mean   : 0.0007662   Mean   : 0.001224
##  3rd Qu.: 0.0073078   3rd Qu.: 0.0078017   3rd Qu.: 0.007377
##  Max.   : 0.0193460   Max.   : 0.0205990   Max.   : 0.028238
##       V46                  V47                  V48
##  Min.   :-0.020269   Min.   :-0.0240489   Min.   :-0.0213995
##  1st Qu.:-0.003419   1st Qu.:-0.0074864   1st Qu.:-0.0066479
##  Median : 0.001195   Median : 0.0002127   Median :-0.0003610
##  Mean   : 0.001438   Mean   : 0.0003668   Mean   :-0.0004886
##  3rd Qu.: 0.008019   3rd Qu.: 0.0065150   3rd Qu.: 0.0076715
##  Max.   : 0.022504   Max.   : 0.0213651   Max.   : 0.0201063
##       V49                  V50
##  Min.   :-0.035272   Min.   :-0.0203089
##  1st Qu.:-0.009119   1st Qu.:-0.0093259
##  Median :-0.001270   Median :-0.0006868
##  Mean   :-0.002874   Mean   :-0.0010836
##  3rd Qu.: 0.004680   3rd Qu.: 0.0061049
##  Max.   : 0.011880   Max.   : 0.0197630
```

(b) Perform PCA on the 60 observations and plot the first two principal component score vectors. Use a different color to indicate the observations in each of the three classes. If the three classes appear separated in this plot, then continue on to part (c). If not, then return to part (a) and modify the simulation so that there is greater separation between the three classes. Do not continue to part (c) until the three classes show at least some separation in the first two principal component score vectors.

```
pca.out = prcomp(x)
summary(pca.out)
```

```
## Importance of components:
```

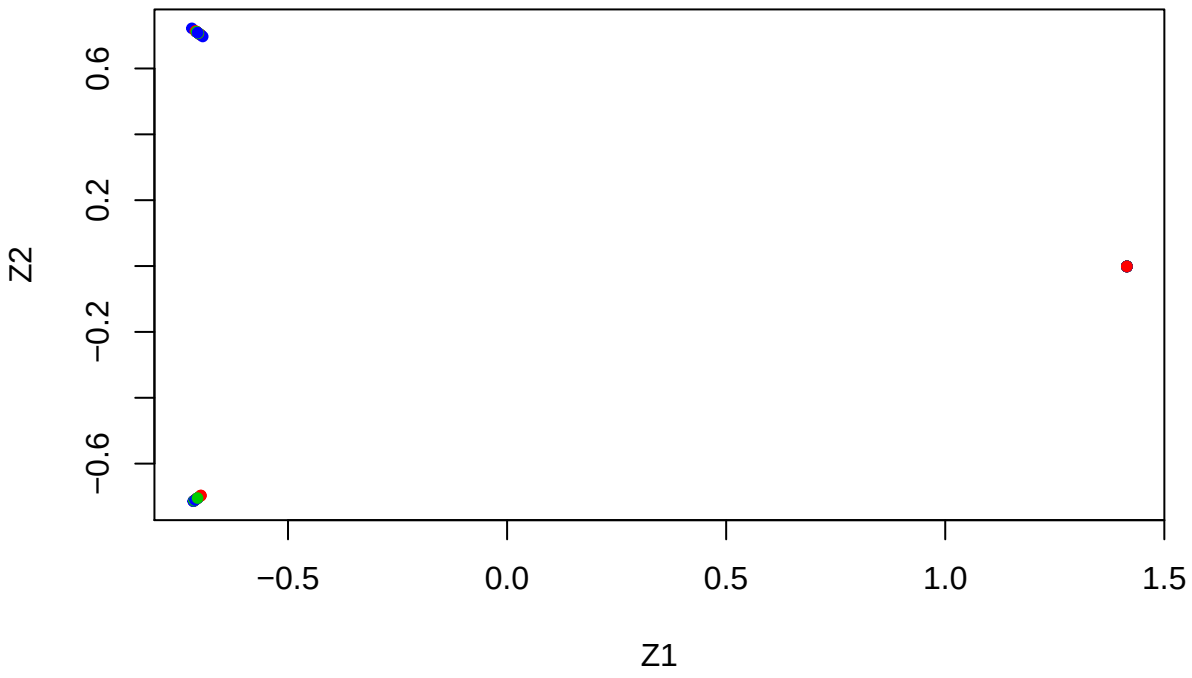
```
##                             PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.0087 0.5826 0.01832 0.01787 0.01678 0.01648 0.01608
## Proportion of Variance 0.7472 0.2493 0.00025 0.00023 0.00021 0.00020 0.00019
## Cumulative Proportion  0.7472 0.9965 0.99677 0.99701 0.99721 0.99741 0.99760
##                               PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.01542 0.01486 0.01382 0.01355 0.01299 0.01283 0.01248
## Proportion of Variance 0.00017 0.00016 0.00014 0.00013 0.00012 0.00012 0.00011
## Cumulative Proportion  0.99778 0.99794 0.99808 0.99822 0.99834 0.99846 0.99857
##                              PC15    PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     0.01208 0.01189 0.01125 0.01115 0.01076 0.01071 0.01017
## Proportion of Variance 0.00011 0.00010 0.00009 0.00009 0.00009 0.00008 0.00008
## Cumulative Proportion  0.99868 0.99879 0.99888 0.99897 0.99906 0.99914 0.99922
##                               PC22     PC23     PC24     PC25     PC26     PC27
## Standard deviation     0.009797 0.009678 0.009471 0.009009 0.008697 0.008027
## Proportion of Variance 0.000070 0.000070 0.000070 0.000060 0.000060 0.000050
## Cumulative Proportion  0.999290 0.999350 0.999420 0.999480 0.999540 0.999580
##                               PC28     PC29     PC30     PC31     PC32     PC33
## Standard deviation     0.007844 0.007417 0.007056 0.006758 0.006541 0.006178
## Proportion of Variance 0.000050 0.000040 0.000040 0.000030 0.000030 0.000030
## Cumulative Proportion  0.999630 0.999670 0.999710 0.999740 0.999770 0.999800
##                               PC34     PC35    PC36     PC37     PC38     PC39
## Standard deviation     0.005903 0.005654 0.00553 0.005241 0.004951 0.004938
## Proportion of Variance 0.000030 0.000020 0.00002 0.000020 0.000020 0.000020
## Cumulative Proportion  0.999820 0.999850 0.99987 0.999890 0.999910 0.999930
##                               PC40     PC41     PC42     PC43     PC44     PC45
## Standard deviation     0.004606 0.004136 0.003893 0.003745 0.003466 0.002827
## Proportion of Variance 0.000020 0.000010 0.000010 0.000010 0.000010 0.000010
## Cumulative Proportion  0.999940 0.999950 0.999960 0.999980 0.999980 0.999990
##                               PC46     PC47     PC48     PC49      PC50
## Standard deviation     0.002304 0.002023 0.001515 0.001086 0.0009141
## Proportion of Variance 0.000000 0.000000 0.000000 0.000000 0.0000000
## Cumulative Proportion  0.999990 1.000000 1.000000 1.000000 1.0000000
```

```
pca.out$x[,1:2]
```

```
##               PC1           PC2
##  [1,] -0.7145255 -0.7126547745
##  [2,] -0.7096651 -0.7080330411
##  [3,] -0.7057317 -0.7041507003
##  [4,] -0.7157223 -0.7139740307
##  [5,] -0.7063183 -0.7043595523
##  [6,] -0.7074823 -0.7054672971
##  [7,] -0.7072340 -0.7052462943
##  [8,] -0.6998180 -0.6978969012
##  [9,] -0.7164180 -0.7142022263
## [10,] -0.6987499 -0.6968434914
## [11,] -0.7128968 -0.7110838193
## [12,] -0.7155857 -0.7137030221
## [13,] -0.7127428 -0.7105238250
## [14,] -0.7059894 -0.7041905658
## [15,] -0.7067596 -0.7049958631
## [16,] -0.7098472 -0.7080456792
## [17,] -0.7144170 -0.7125933044
## [18,] -0.7122645 -0.7106223571
```

```
## [19,] -0.6991049 -0.6969386522
## [20,] -0.7065192 -0.7045804818
## [21,]  1.4144704 -0.0012468897
## [22,]  1.4146667 -0.0009748514
## [23,]  1.4146314 -0.0010643164
## [24,]  1.4147095 -0.0009415425
## [25,]  1.4146024 -0.0009443586
## [26,]  1.4142893 -0.0011852496
## [27,]  1.4144636 -0.0009184244
## [28,]  1.4146406 -0.0009300176
## [29,]  1.4145193 -0.0010596937
## [30,]  1.4144932 -0.0011468907
## [31,]  1.4145345 -0.0013050921
## [32,]  1.4144790 -0.0012649359
## [33,]  1.4146083 -0.0010955647
## [34,]  1.4143606 -0.0009461704
## [35,]  1.4145507 -0.0009567615
## [36,]  1.4144521 -0.0007408849
## [37,]  1.4144438 -0.0012769954
## [38,]  1.4144898 -0.0011498007
## [39,]  1.4145092 -0.0014299226
## [40,]  1.4144943 -0.0011576340
## [41,] -0.7017562  0.7041360193
## [42,] -0.7090311  0.7113365007
## [43,] -0.6986115  0.7012362079
## [44,] -0.7017243  0.7044538356
## [45,] -0.6988853  0.7014346181
## [46,] -0.7017244  0.7042086143
## [47,] -0.7046426  0.7071311182
## [48,] -0.7195172  0.7218444853
## [49,] -0.7109993  0.7134658470
## [50,] -0.7027761  0.7053248878
## [51,] -0.7093148  0.7116805050
## [52,] -0.7080713  0.7105732150
## [53,] -0.7089515  0.7112381089
## [54,] -0.6946658  0.6970857928
## [55,] -0.7110987  0.7134945803
## [56,] -0.7091710  0.7116503287
## [57,] -0.7060741  0.7086489854
## [58,] -0.7044186  0.7071207644
## [59,] -0.7048852  0.7071477599
## [60,] -0.7062974  0.7086297015
```

```
plot(pca.out$x[,1:2], col = 2:4, xlab = 'Z1', ylab = 'Z2', pch = 20)
```



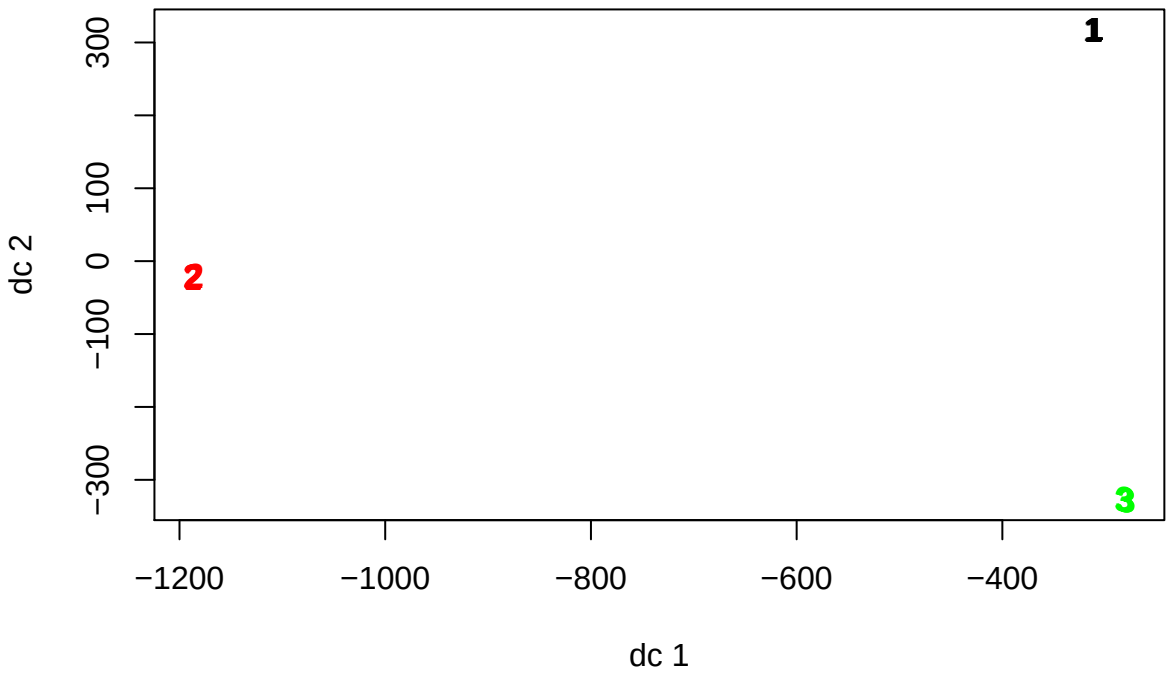
(c) Perform K-means clustering of the observations with K = 3. How well do the clusters that you obtained in K-means clustering compare to the true class labels?

Hint: You can use the `table()` function in R to compare the true class labels to the class labels obtained by clustering. Be careful how you interpret the results: K-means clustering will arbitrarily number the clusters, so you cannot simply check whether the true class labels and clustering labels are the same.

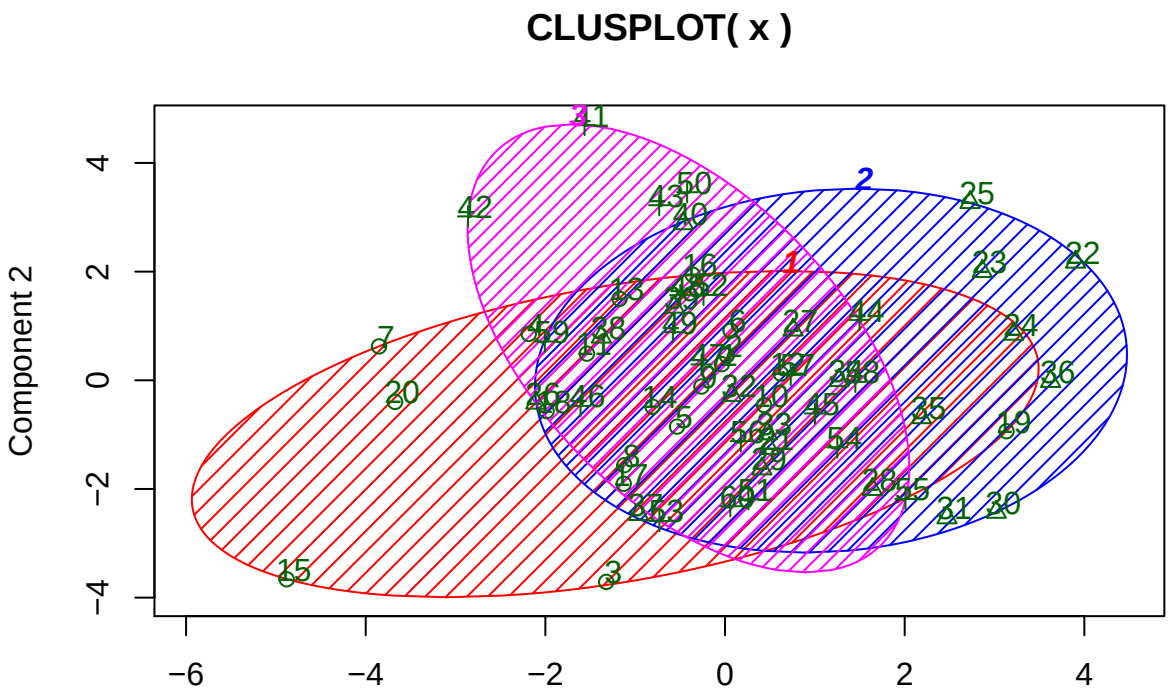
```
library(cluster)
library(fpc)
kmeans.out.3 = kmeans(x, 3, nstart = 20)
table(true.labels, kmeans.out.3$cluster)
```

```
##
## true.labels  1  2  3
##           1 20  0  0
##           2  0 20  0
##           3  0  0 20
```

```
plotcluster(x, kmeans.out.3$cluster)
```




```
clusplot(x, kmeans.out.3$cluster, color = TRUE, shade = TRUE, labels = 2, lines = 0)
```



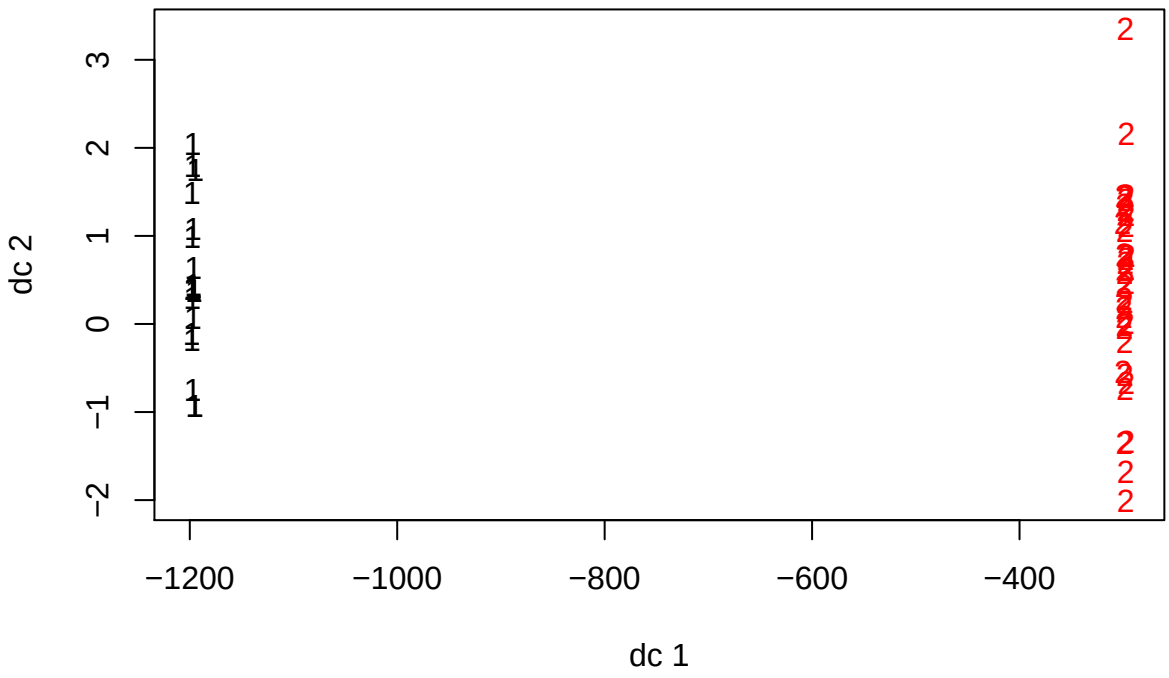
There are three clusters defined in the dataset with 20 observation each. They are perfectly clustered in three classes from the `kmeans()` clustering. In the plot you can see there are three clusters numbered 1,2 and 3.

(d) Perform K-means clustering with K = 2. Describe your results.

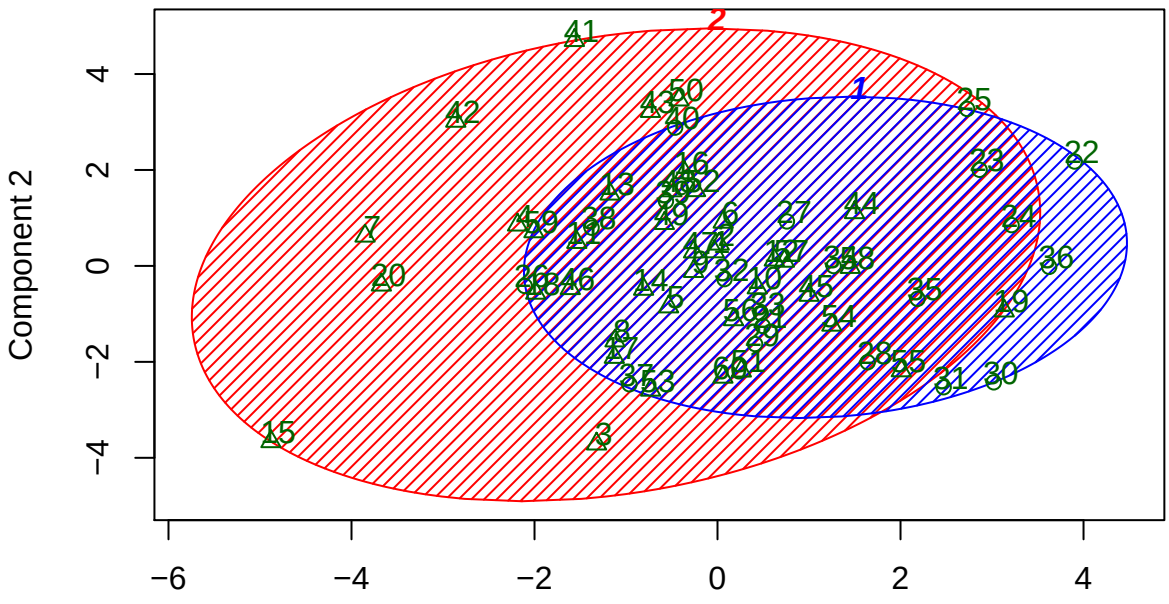
```
kmeans.out.2 = kmeans(x, 2, nstart = 20)
table(true.labels, kmeans.out.2$cluster)
```

```
##
## true.labels  1  2
##           1  0 20
##           2 20  0
##           3  0 20
```

```
plotcluster(x, kmeans.out.2$cluster)
```




```
clusplot(x, kmeans.out.2$cluster, color = TRUE, shade = TRUE, labels = 2, lines = 0)
```



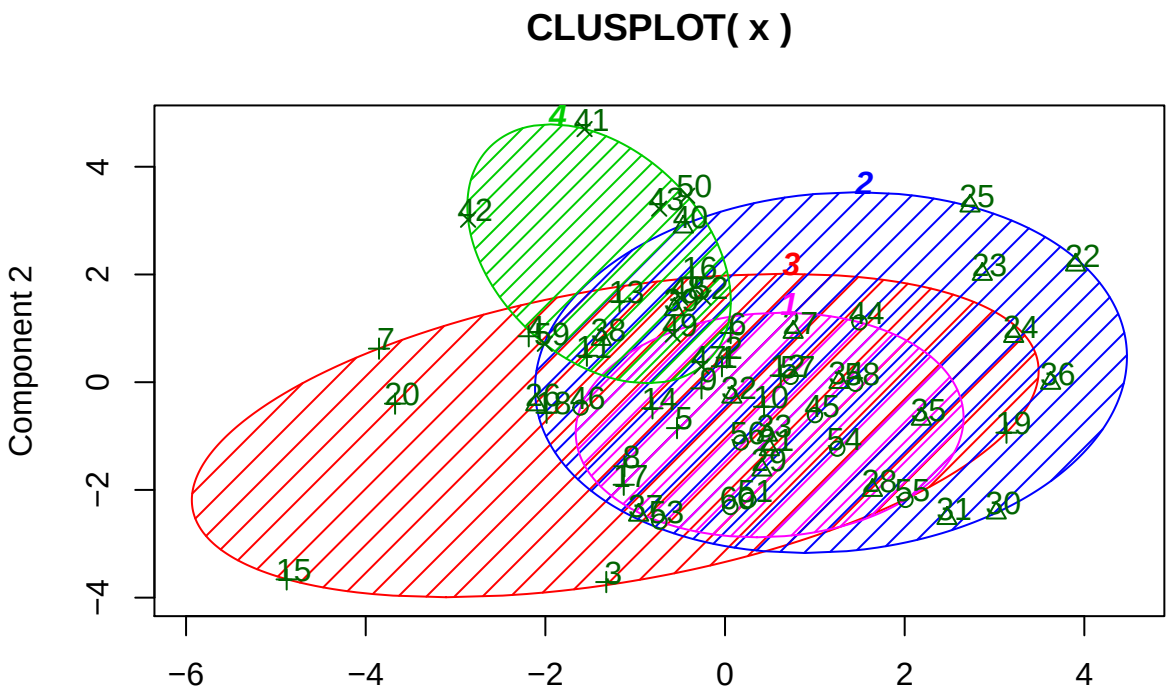
While we define $\mathbf{k = 2}$. The two clusters point merge into one cluster. As you can see in the plot, the cluster 1 and 2 from previous plot merged into 1 cluster, Leaving **20** observations in one and **40** observations in the other cluster.

(e) Now perform K-means clustering with K = 4, and describe your results.

```
kmeans.out.4 = kmeans(x, 4, nstart = 20)
table(true.labels, kmeans.out.4$cluster)
```

```
##
## true.labels  1  2  3  4
##           1  0  0 20  0
##           2  0 20  0  0
##           3 11  0  0  9
```

```
clusplot(x, kmeans.out.4$cluster, color = TRUE, shade = TRUE, labels = 2, lines = 0)
```



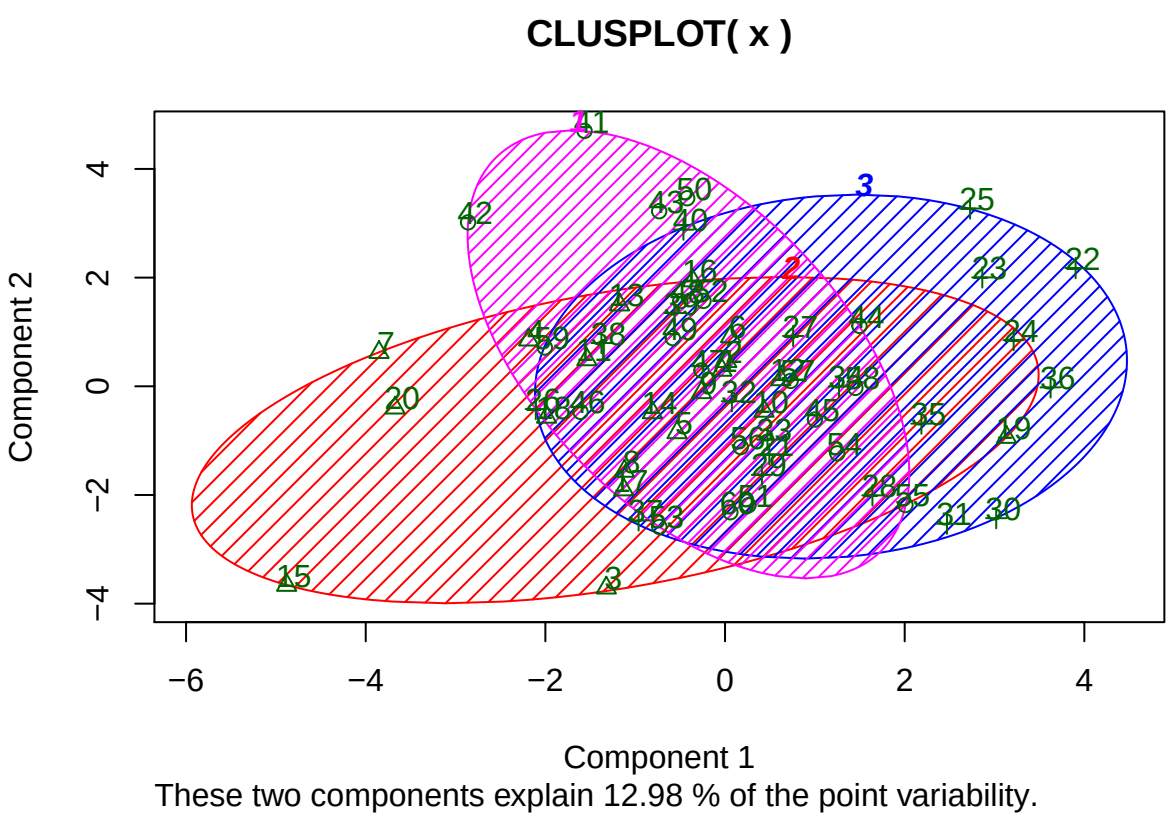
The defined dataset with 3 classes are now classified into 4 clusters. With **cluster-2** and **cluster-3** perfectly clustered with **20** observartion each and other two contains the distributed observations of third one, with **9** in **cluster-4** and **11** in **cluster-1**.

(f) Now perform K-means clustering with K = 3 on the first two principal component score vectors, rather than on the raw data. That is, perform K-means clustering on the 60 × 2 matrix of which the first column is the first principal component score vector, and the second column is the second principal component score vector. Comment on the results.

```
kmeans.out.31 = kmeans(pca.out$x[,1:2], 3, nstart=20)
table(true.labels, kmeans.out.31$cluster)
```

```
##
## true.labels  1  2  3
##           1  0 20  0
##           2  0  0 20
##           3 20  0  0
```

```
clusplot(x, kmeans.out.31$cluster, color = TRUE, shade = TRUE, labels = 2, lines = 0)
```



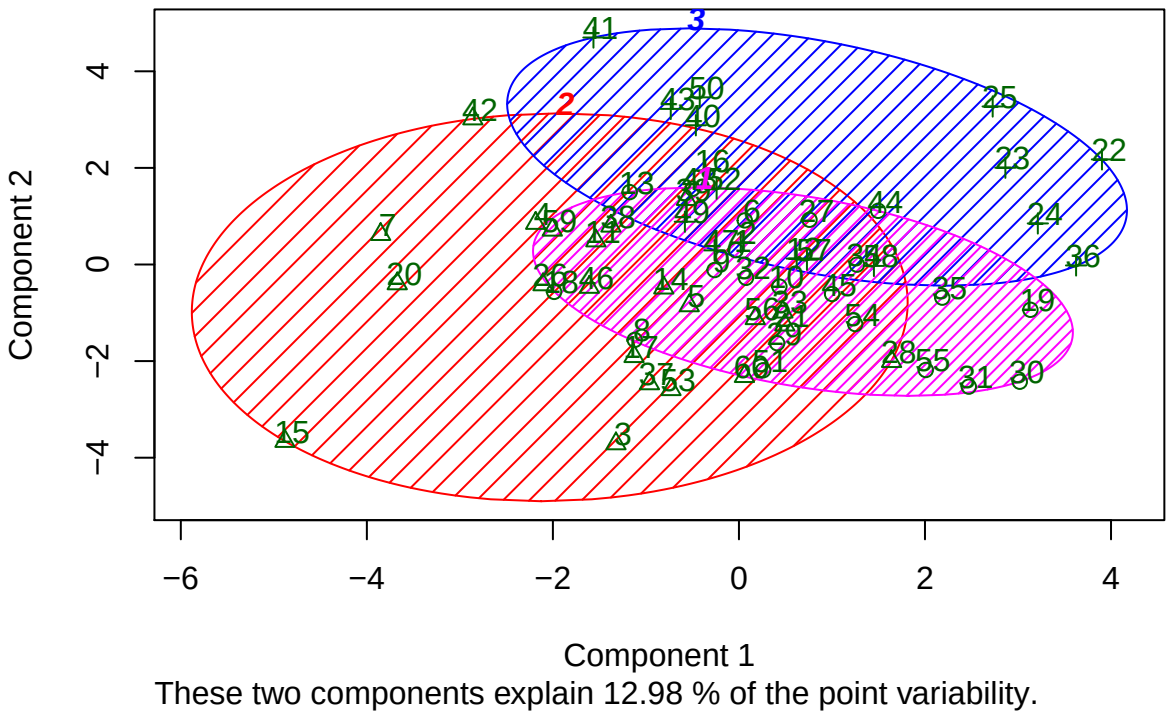
The results show the correctly classified instances into three clusters, with **20** observation each. Perfectly clustered again.

(g) Using the `scale()` function, perform K-means clustering with K = 3 on the data after scaling each variable to have standard deviation one. How do these results compare to those obtained in (b)? Explain.

```
kmeans.scale<-kmeans(scale(x),3,nstart=20)
table(true.labels,kmeans.scale$cluster)
```

```
##
## true.labels 1 2 3
##           1 8 9 3
##           2 7 7 6
##           3 6 6 8
```

```
clusplot(x, kmeans.scale$cluster, color = TRUE, shade = TRUE, labels = 2, lines = 0)
```



Here in the truth table we can see that the observations are not perfectly clustered. The results are not good compared to the unscaled clustering(b). The `scale()` function affects the **distances** between the observations.