Assignment-1

Tom Qu

2/24/2019

Q1

(a)

```
library(gclus)
```

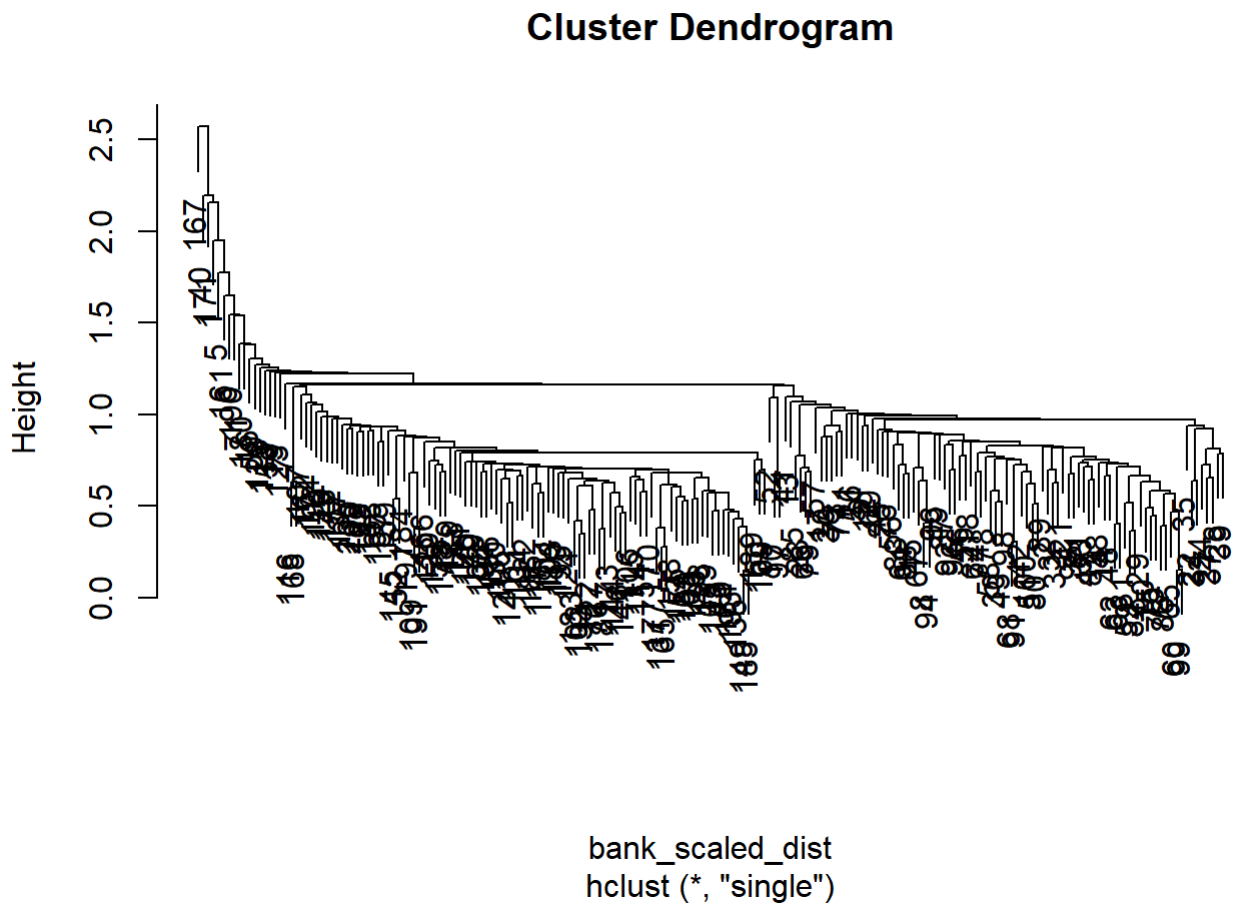
```
## Loading required package: cluster
```

```
data(bank)
```

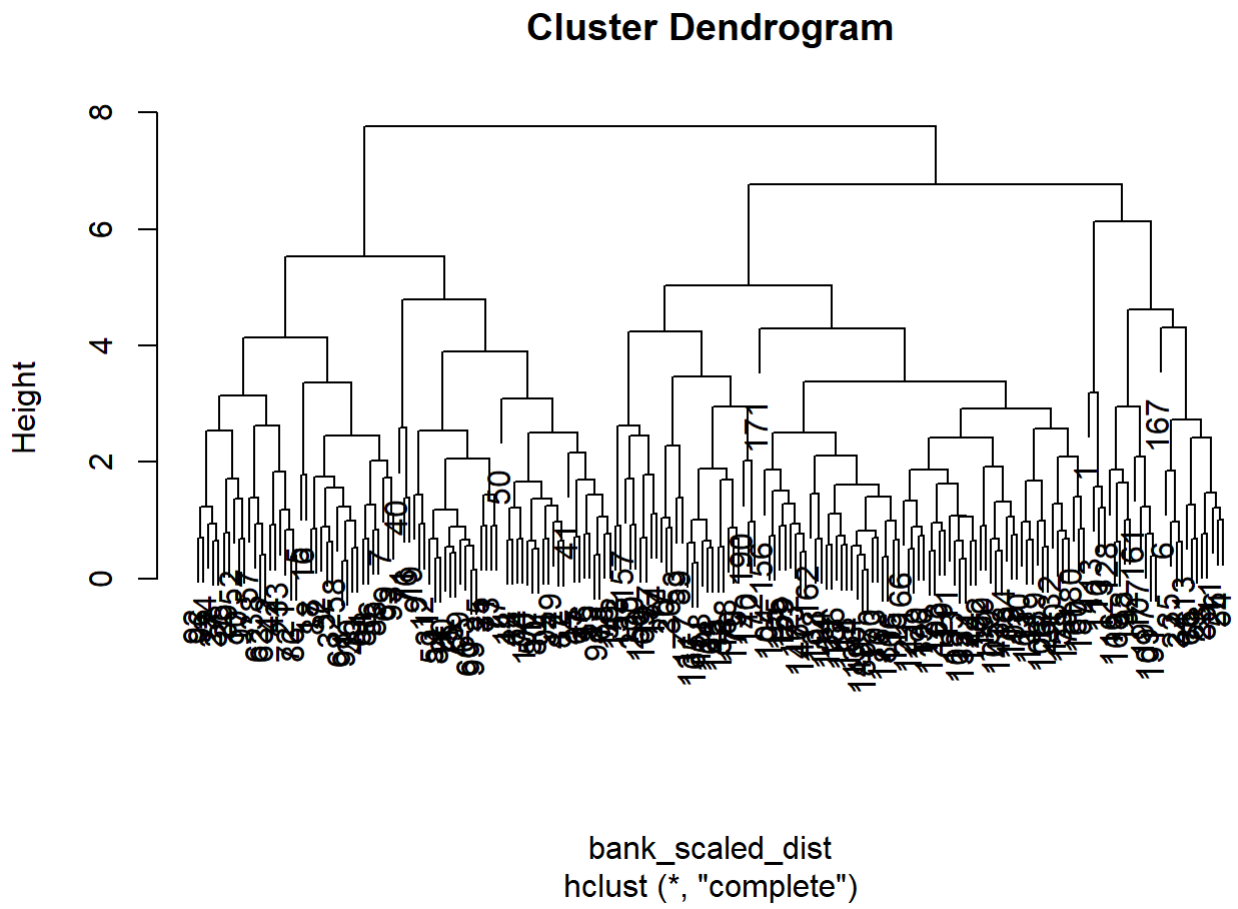
We should consider one of the standardized distance measures (standardized or Mahalanobis), as even though the measuring units are the same across each bill measurement, the scale is quite different across them (Diagonal has a standard deviation > 4 times Length or Left, for instance).

(b)

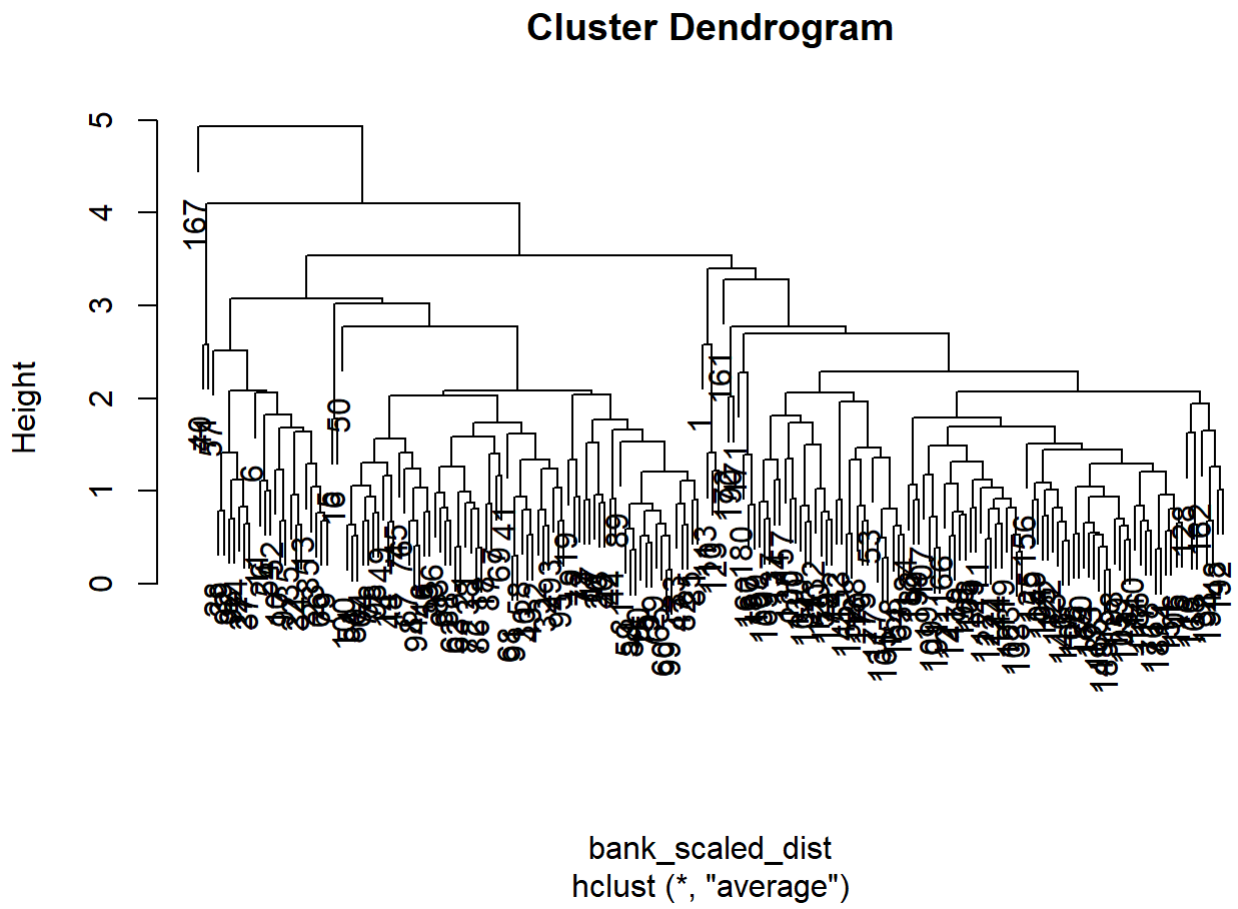
```
bank_scaled_dist <- dist(scale(bank[,-c(1,5)]))
plot(hclust(bank_scaled_dist, method="single"))
```




```
plot(hclust(bank_scaled_dist, method="complete"))
```




```
plot(hclust(bank_scaled_dist, method="average"))
```



(c)

The complete linkage dendrogram appears to give the clearest group structure. Both single linkage and average linkage will put several observations in their own group.

(d)

The dendogram suggests a choice of K=2.

```
hcres <- cutree(hclust(bank_scaled_dist, method="complete"), 2)
table(bank$Status, hcres)
```

```
##    hcres
##       1   2
##   0  19  81
##   1 100   0
```

```
19/200 #misclassification rate
```

```
## [1] 0.095
```

We misclassify 19 genuine notes into the group that is made of primarily counterfeit notes, for a misclassification rate of 0.095.

(e)

```
set.seed(632)
kscale <- kmeans(bank_scaled_dist, 2)
table(bank$Status, kscale$clus)
```

```
##
##      1  2
##   0 20 80
##   1 98  2
```

```
22/200 #misclassification rate
```

```
## [1] 0.11
```

For this run of kmeans, we misclassify 20 genuine notes into the group that is made of primarily counterfeit notes, along with 2 counterfeits in the primarily genuine group, for a misclassification rate of 0.11.

(f)

```
set.seed(632)
kraw <- kmeans(bank[,-c(1,5)], 2)
table(bank$Status, kraw$clus)
```

```
##
##       1   2
##   0   1  99
##   1 100   0
```

```
1/200 #misclassification rate
```

```
## [1] 0.005
```

For this run of kmeans on the raw data, we misclassify only one of the bank notes for a rate of 0.005! This actually goes **against** our generalized rules discussed in class. Since we are looking at the raw data, predictors with high variability will dominate while building groups. It happens, in this case, that the higher variability predictor (Diagonal) actually provides a clearer picture of the two groups than the lower variability predictors (such as Length and Left). Hence, this is a case where standardization will actually hurt us in finding the known group structure within the data.

(g)

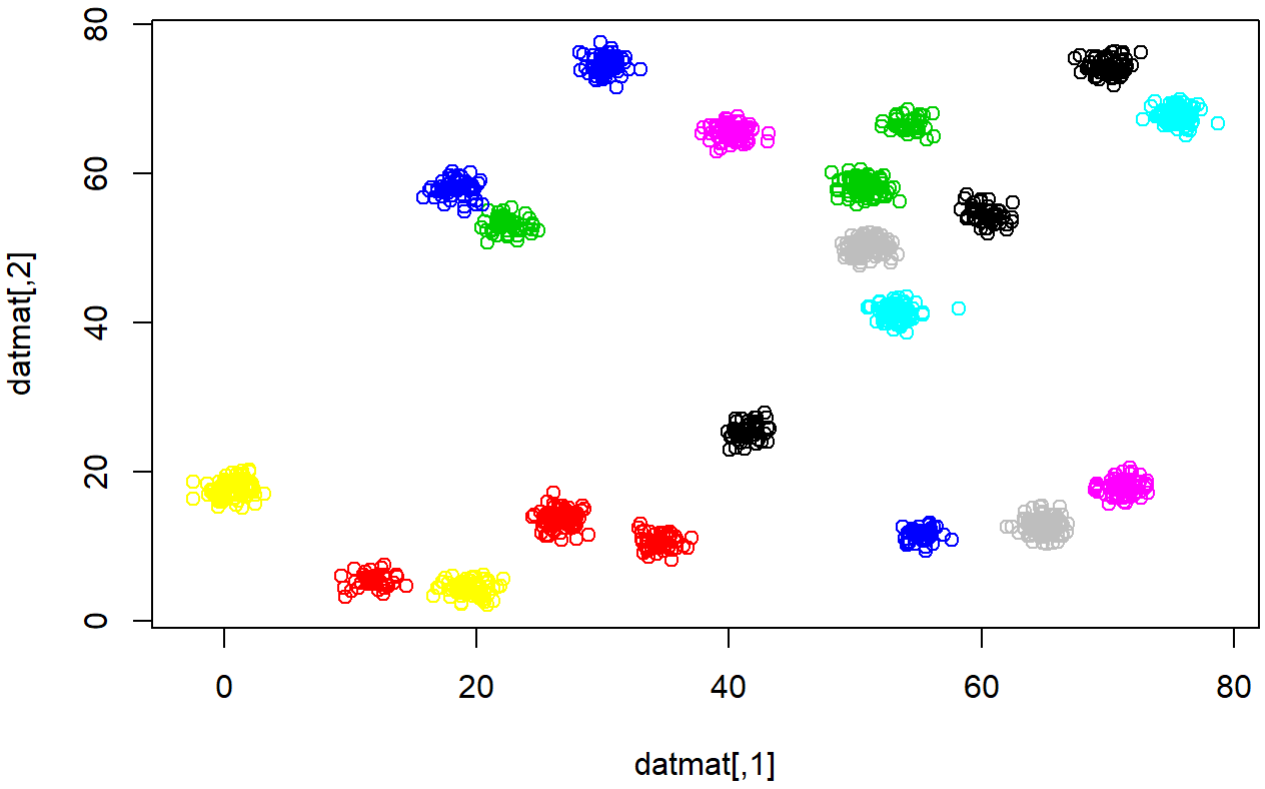
Overall, the strong performance of the clustering algorithms tells us that there is a very strong group structure in the data — which actually corresponds to the known categories of counterfeit and genuine bills.

Q2

```
load("C:/Users/yizhe/Desktop/MDS/Term5/data_573/data/lots.Rdata")
```

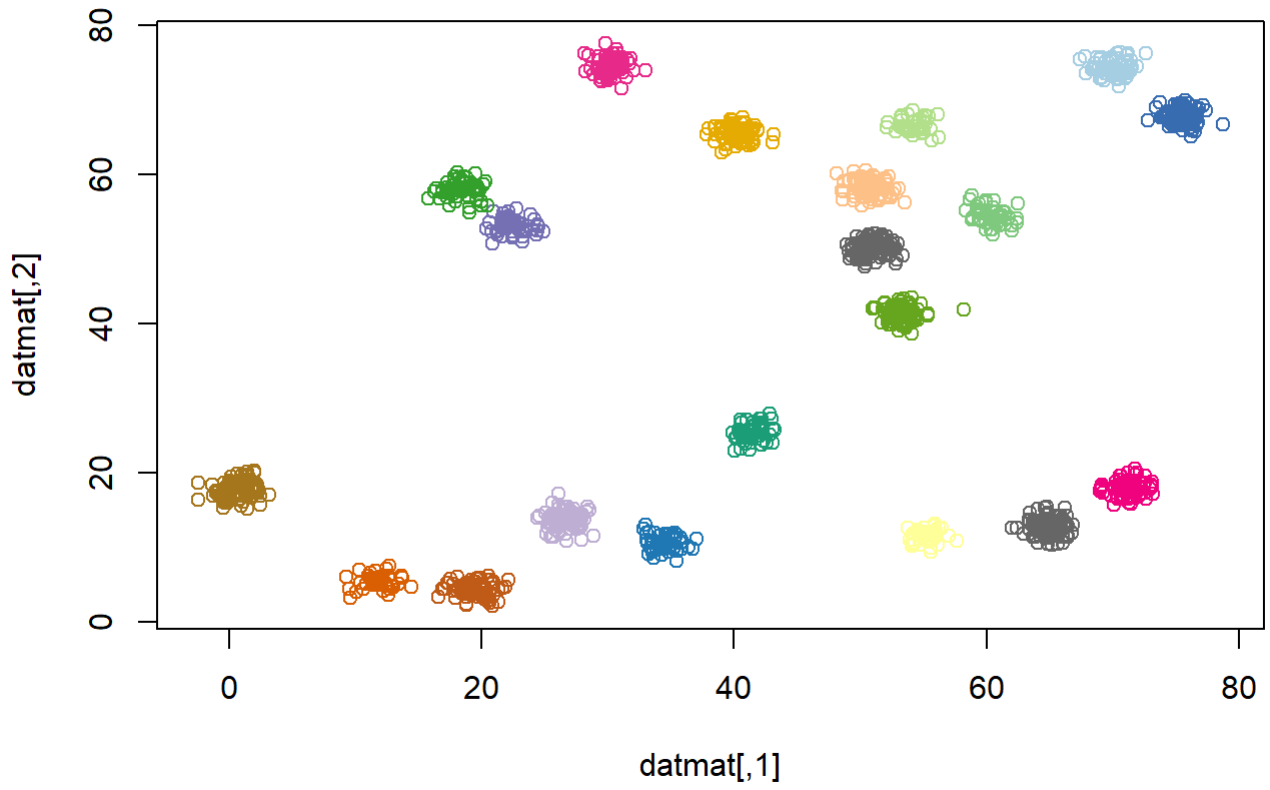
(a)

```
plot(datmat, col=clusts)
```



Note that the above plot is pretty useless since R is recycling the colour vector. Here's an improvement…

```
library(RColorBrewer)
qual_col_pals = brewer.pal.info[brewer.pal.info$category == 'qual',]
col_vector = unlist(mapply(brewer.pal, qual_col_pals$maxcolors, rownames(qual_col_pals)))
palette(col_vector)
plot(datmat, col=clusts)
```



(b)

```
library(mclust)
```

```
## Warning: package 'mclust' was built under R version 3.5.2
```

```
## Package 'mclust' version 5.4.2
## Type 'citation("mclust")' for citing this R package in publications.
```

```
set.seed(461)
r1 <- kmeans(datmat, 20)
adjustedRandIndex(clusts, r1$cluster)
```

```
## [1] 0.8317588
```

(c)

```
set.seed(41)
r2 <- kmeans(datmat, 20)
adjustedRandIndex(clusts, r2$cluster)
```

```
## [1] 0.6747311
```

(d)

```
set.seed(461)
r3 <- kmeans(datmat, 20, nstart=1000)
adjustedRandIndex(clusts, r3$cluster)
```

```
## [1] 0.9438239
```

(e)

```
set.seed(41)
r4 <- kmeans(datmat, 20, nstart=1000)
adjustedRandIndex(clusts, r4$cluster)
```

```
## [1] 1
```

(f)

Each run provides a different result with respect to classification of the true groups, which is surprising as the groups structures are relatively clear (as seen in part ‘a’). We could also tell this from viewing the within group sum of squares for each:

```
r1$tot.withinss
```

```
## [1] 14039.05
```

```
r2$tot.withinss
```

```
## [1] 19033.38
```

```
r3$tot.withinss
```

```
## [1] 4010.399
```

```
r4$tot.withinss
```

```
## [1] 2933.074
```

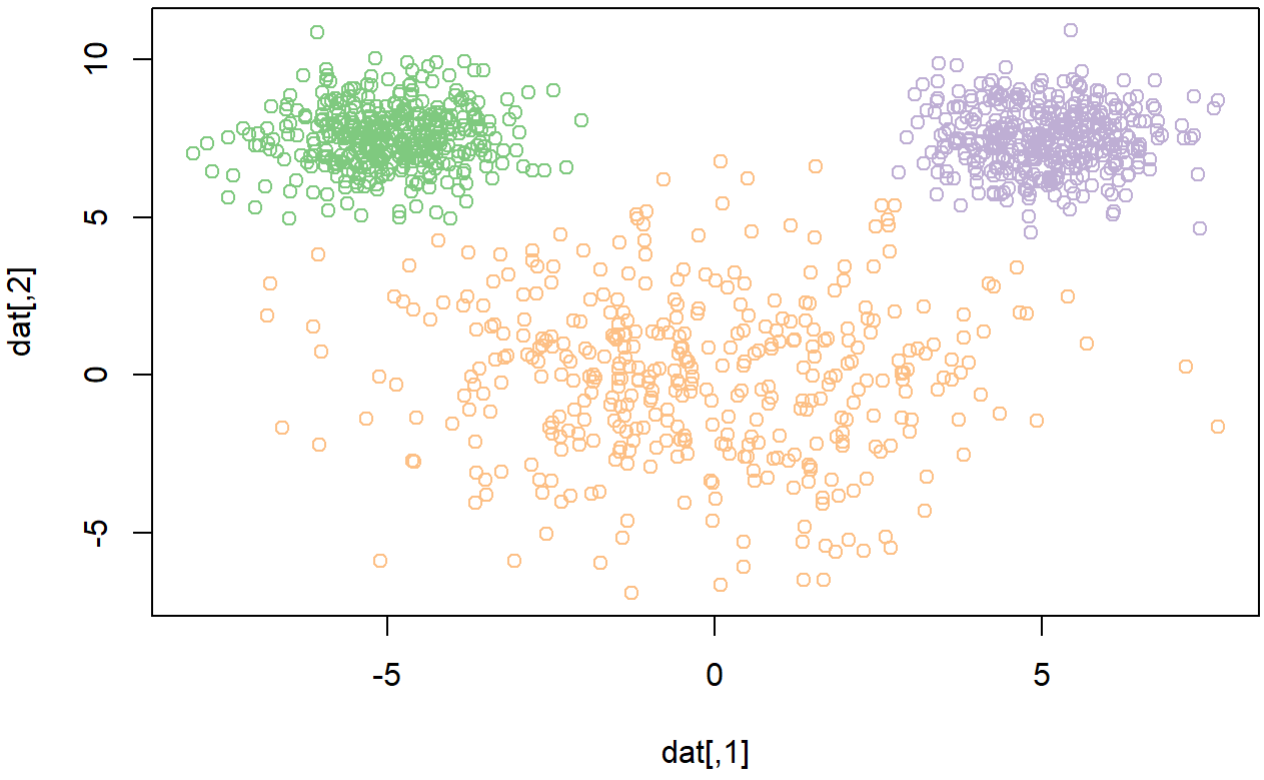
It's worth noting that the runs with many random starts (`r3` and `r4`) have significantly lower within group sum of squares than the individual runs (`r1` and `r2`). But while `r4` does achieve perfect classification, `r3` does not — and this is with 1000 random starts. All of this illustrates concerns with k-means being able to consistently find the global minima for data with this many groups.

Q3

```
library(mvtnorm)
```

```
## Warning: package 'mvtnorm' was built under R version 3.5.2
```

```
set.seed(35151)
le <- rmvnorm(400, mean = c(-5,7.5))
re <- rmvnorm(400, mean = c(5,7.5))
hd <- rmvnorm(400, mean = c(0,0), sigma=7*diag(2) )
dat <- rbind(le, re, hd)
mrun <- Mclust(dat)
plot(dat, col=mrun$class)
```




```
summary(mrun)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VII (spherical, varying volume) model with 3 components:
##
##  log.likelihood    n df       BIC       ICL
##       -5396.482 1200 11 -10870.95 -10881.98
##
## Clustering table:
##   1   2   3
## 401 399 400
```

```
table(rep(c(1,2,3), each=400), mrun$class)
```

```
##
##       1   2   3
##   1 400   0   0
##   2   0 397   3
##   3   1   2 397
```

Yes, the result is more sensible. The VII model is chosen, which allows the volume (essentially, group size) to differ among groups — this is important since the “ears” are smaller than the “head”. From the table provided, one can see that only 6 observations total are misclassified.

Q4

```
load("C:/Users/yizhe/Desktop/MDS/Term5/data_573/data/asim.Rdata")
x <- asim[,-1]
y <- asim[, 1]
u1 <- hclust(dist(scale(x)))
res1 <- cutree(u1, 2)
ulinmod2 <- lm(y~x*res1)
summary(ulinmod2)
```

```
##
## Call:
## lm(formula = y ~ x * res1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.99101 -0.59457  0.03553  0.61065  2.98516
##
## Coefficients:
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -5.31013    2.70073   -1.966   0.0499 *
## x            7.62656    0.08841   86.261  < 2e-16 ***
## x           -1.72147    0.05798  -29.689  < 2e-16 ***
## x           -6.59587    0.06230 -105.868  < 2e-16 ***
## x           -2.07703    0.04572  -45.433  < 2e-16 ***
## x            2.08090    0.06557   31.733  < 2e-16 ***
## x            0.11227    0.07939    1.414   0.1580
## x           -1.23255    0.10343  -11.917  < 2e-16 ***
## x           -7.15220    0.08045  -88.907  < 2e-16 ***
## x            2.59536    0.06502   39.914  < 2e-16 ***
## res1         7.56585    1.76602    4.284 2.22e-05 ***
## x:res1      -5.11942    0.07260  -70.516  < 2e-16 ***
## x:res1       1.90348    0.04313   44.138  < 2e-16 ***
## x:res1       4.27634    0.04067  105.159  < 2e-16 ***
## x:res1      -0.02066    0.03426   -0.603   0.5468
## x:res1      -0.43149    0.04341   -9.940  < 2e-16 ***
## x:res1       0.89608    0.05052   17.737  < 2e-16 ***
## x:res1      -0.73147    0.07931   -9.223  < 2e-16 ***
## x:res1       4.95254    0.05917   83.704  < 2e-16 ***
## x:res1      -0.23190    0.04657   -4.979 8.92e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9774 on 480 degrees of freedom
## Multiple R-squared:  0.9956, Adjusted R-squared:  0.9955
## F-statistic:  5773 on 19 and 480 DF,  p-value: < 2.2e-16
```

Pretty much any clustering algorithm can be used to ‘generate’ the new categorical predictor, as the group structure is very strong. Utilizing that predictor in the linear model as an interactor with all the other variables provides the model that surpasses the 0.99 threshold for R^2.