Lab10

Channing_503128649

15/10/2021

The Data Set

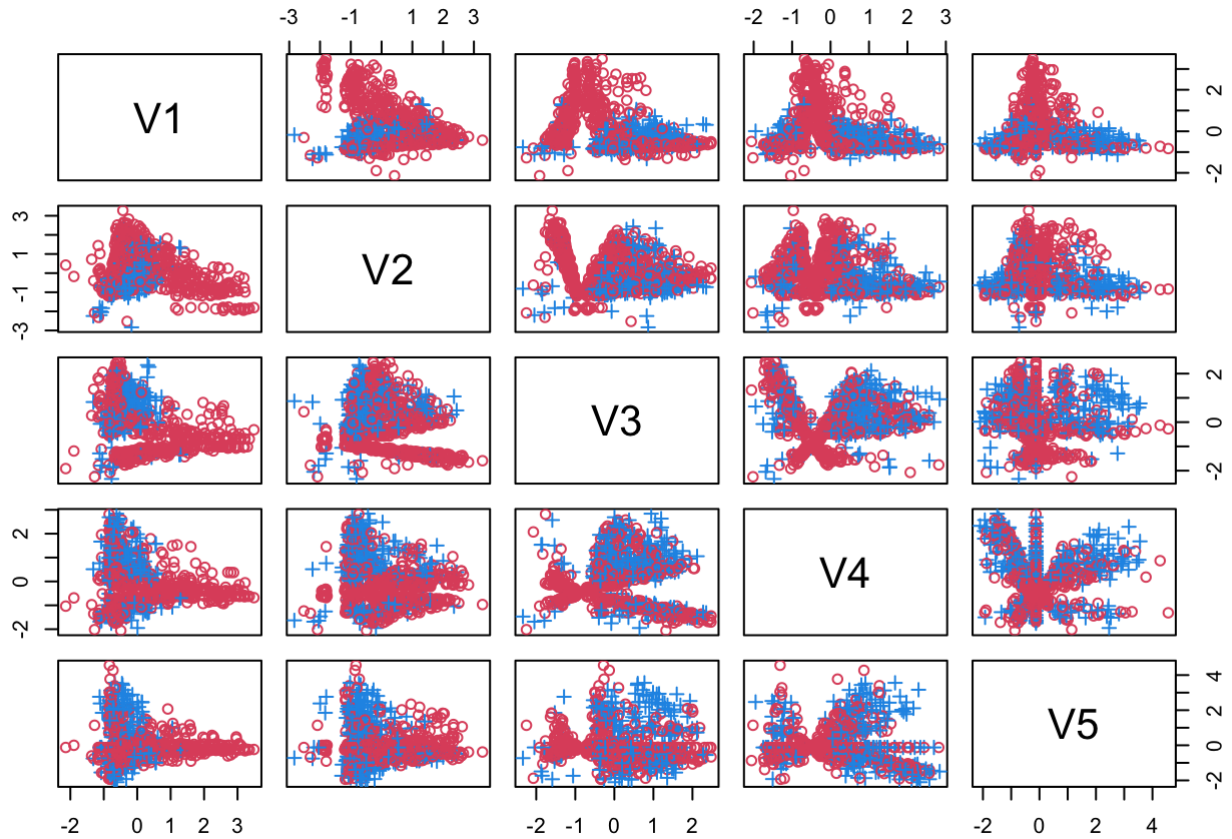
The original Phoneme data set has 5404 observations. To make computation a bit faster for this assignment, I have downsized it to 1000 observations via a random subsampling, and the downsized data set is available on Canvas. You can save a local copy and read it into R as follows:

```
phoneme = read.csv("phoneme.csv", stringsAsFactors=TRUE)
head(phoneme)
```

```
##        V1      V2      V3      V4      V5 Class
## 1  1.7494 -0.0504 -0.4321  0.7681 -0.5974 Nasal
## 2  1.9682 -0.8207 -0.2183  0.0900  0.4641 Nasal
## 3 -0.1535 -0.1977  1.1861  0.1394 -0.6372  Oral
## 4 -0.3297 -0.2271  0.3609 -1.5044 -1.9235  Oral
## 5  2.6405 -0.7292 -1.0152 -0.3963 -0.2138 Nasal
## 6 -0.3896 -0.1427  1.1188  0.7823  1.6400  Oral
```

The data set has 5 numeric predictors and one response variable class, which has two levels: Nasal (nasal vowels) and Oral (oral vowels). The numerical variables have all be standardised (and followed with a rounding to 4 d.p.) to have mean 0 and standard deviation 1, as also in the original data set. A pairwise scatterplot of the data looks like the following:

```
with(phoneme, pairs(phoneme[,-6], col=c(2,4)[Class], pch=c(1,3)[Class]))
```



To perform the following tasks, make sure you have loaded the following library:

```
library(mclust)
```

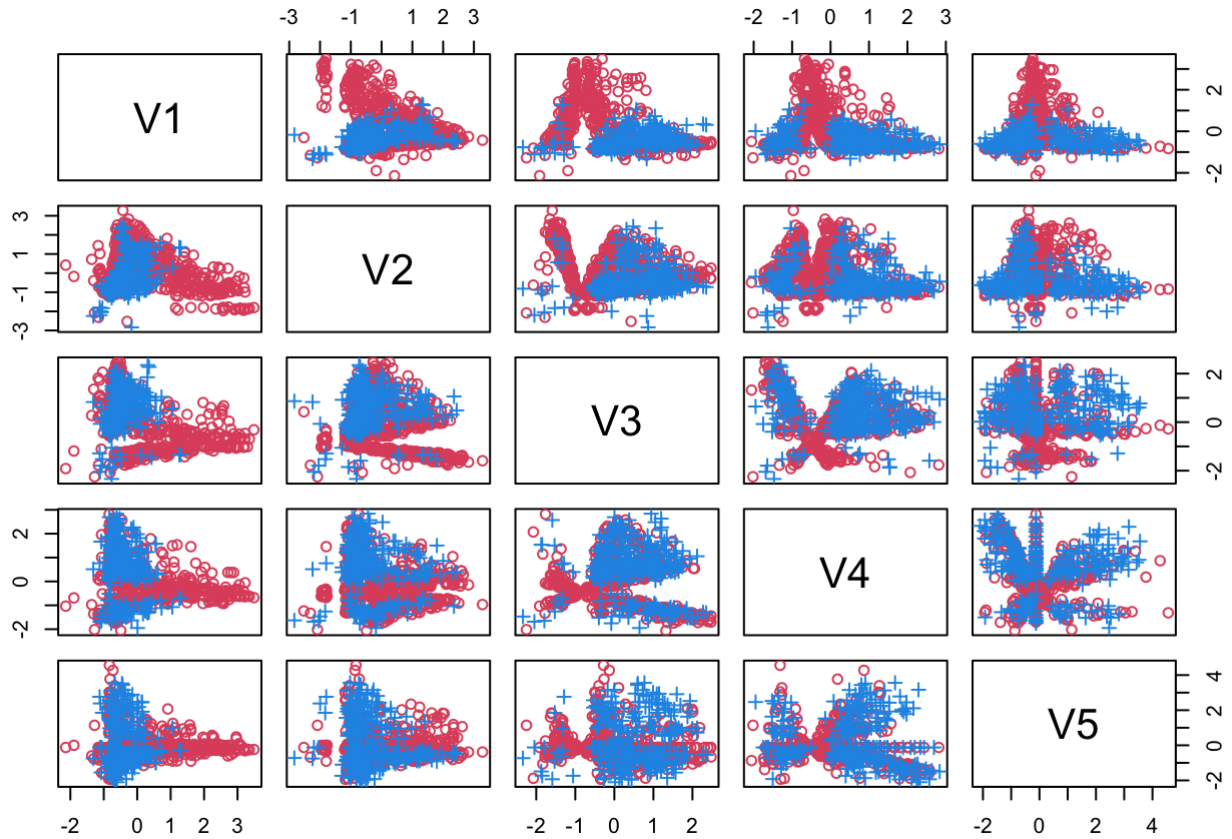
```
## Package 'mclust' version 5.4.7
## Type 'citation("mclust")' for citing this R package in publications.
```

Tasks

Visualisation

1. When there are a large number of observations, observations of smaller classes may be overwhelmed in a scatterplot by those in larger classes. Therefore, one may prefer to plot the observations of larger classes first and then those of smaller ones, or for a two-class data set those of the majority class and then those of the minority one. Write R rode to reproduce the following plot which follows the above description. Make sure that the majority class is determined by code automatically (i.e., not by eyes) so it can be reused easily.

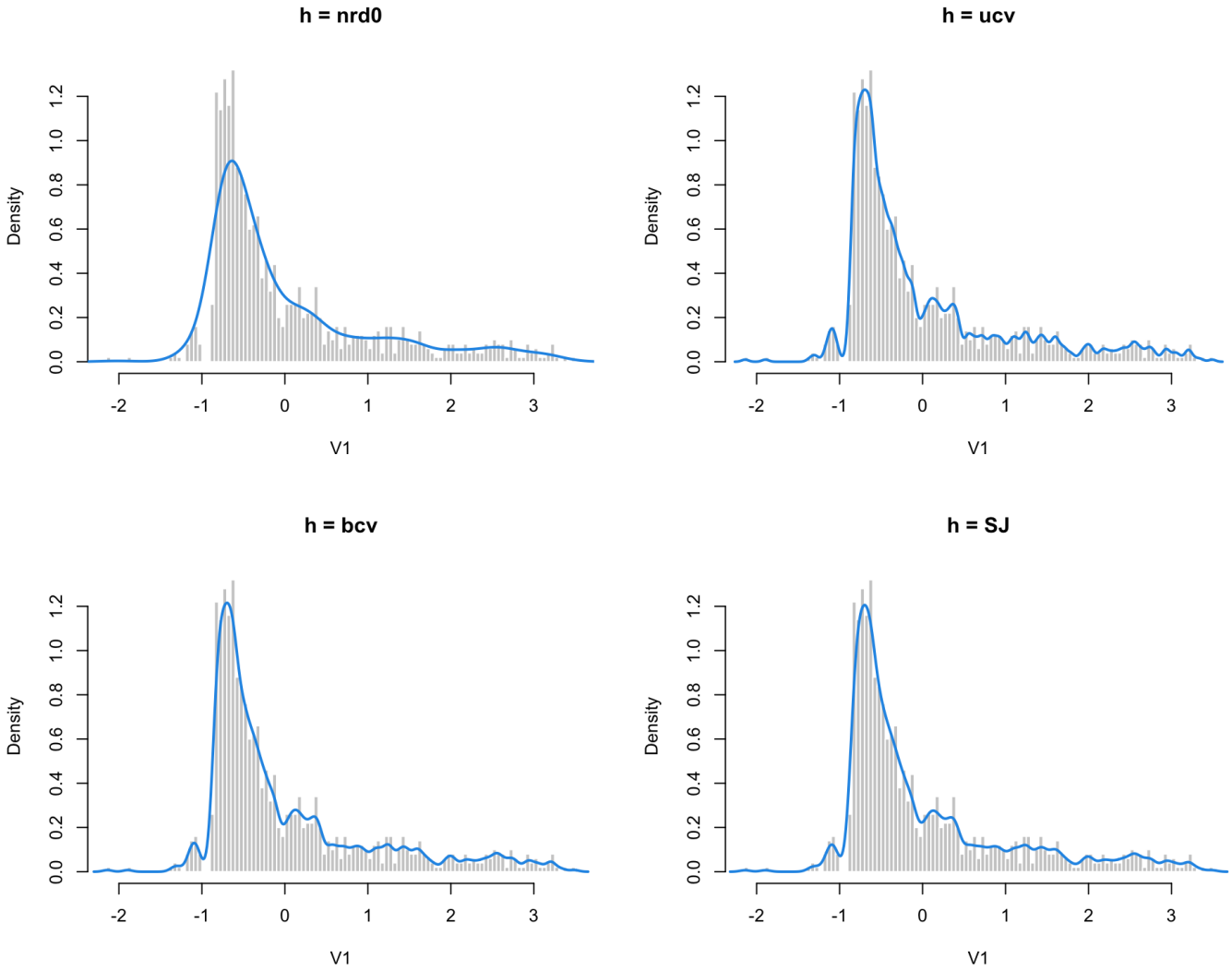
```
phoneme <- if(nrow(phoneme[phoneme$Class==phoneme$Class[1],]) > nrow(phoneme)/2){
  phoneme <- phoneme[order(phoneme$Class),]
} else phoneme[order(phoneme$Class, decreasing=TRUE),]

with(phoneme, pairs(phoneme[,-6], col=c(2,4)[Class], pch=c(1,3)[Class]))
```



Univariate Density Estimation

2. Plot each of the kernel density estimate, by superimposing it on a histogram (with breaks=100), for V1, with the bandwidth values chosen by methods nrd0, ucv, bcv and SJ, respectively. Observe and discuss if both overfitting and underfitting can exist for a KDE.

```
V1 <- phoneme$V1

par(mfrow=c(2,2))
for(bw in c("nrd0", "ucv", "bcv", "SJ")) {
   hist(V1, freq=FALSE, breaks=100,
        col="gray80", border="white",
        main=paste0("h = ", bw), xlab="V1")
   lines(density(V1, bw=bw), col=4, lwd=2)
}
```



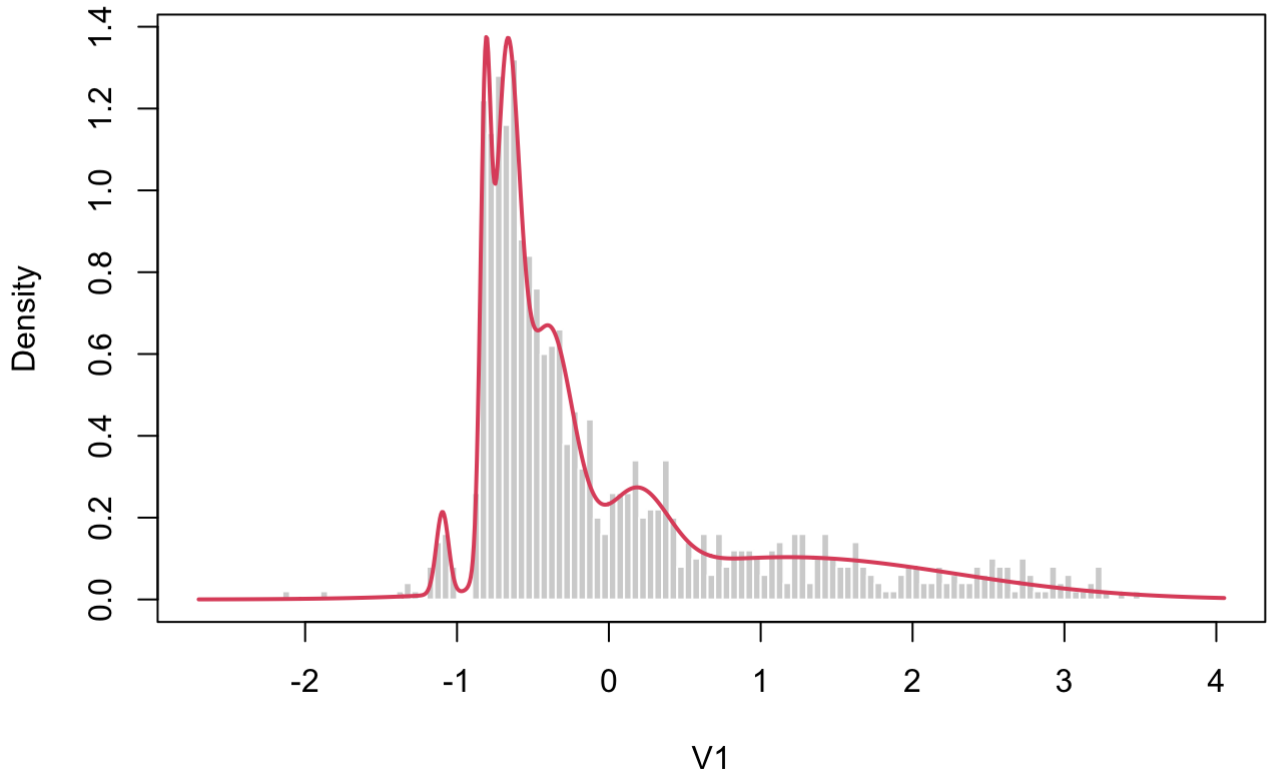
"nrd0" underfit the data vs. "ucv" overfit the data. "bcv" and "SJ" had similar performance, being better matches for the data.

3. Find the best normal mixture fit to V1 (according to the BIC), in both equal and varying variance subfamilies, with the number of components ranging from 1 to 20. Plot the fitted density, along with a histogram of the data.Does it look like a better or worse fit than the best of the KDEs?

```
(best <- densityMclust(V1, G=1:20))
```

```
## 'densityMclust' model object: (V,6)
##
## Available components:
##  [1] "call"           "data"           "modelName"      "n"
##  [5] "d"              "G"              "BIC"            "loglik"
##  [9] "df"             "bic"            "icl"            "hypvol"
## [13] "parameters"     "z"              "classification" "uncertainty"
## [17] "density"
```

```
r = densityMclust(V1, G=6, modelNames="V")
plot(r, V1, "density", breaks=100, lwd=2, col=2)
```



It's a better fit as it followed the trend at the top, and much smoother at the bottom (not overfitting).

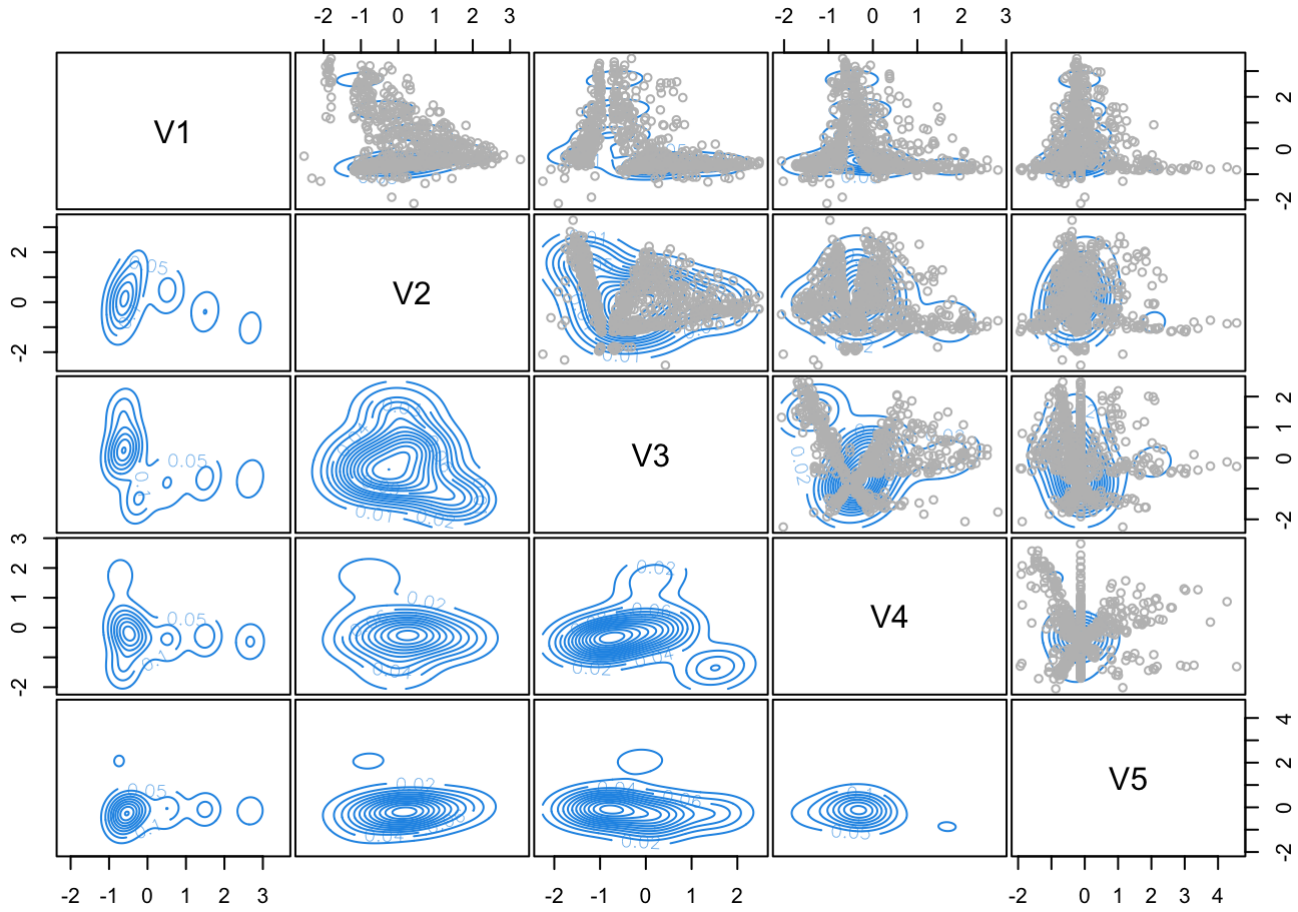
Multivariate density estimation

4. For each of the two classes, find a density estimate in the equal variance subfamily of multivariate normal mixtures, with the number of components ranging from 1 to 9. Show each density in a pairwise plot of the data.

```
(r1e <- densityMclust(phoneme[phoneme$Class=="Nasal",-6], G=1:9, modelNames="EEE"))
```

```
## 'densityMclust' model object: (EEE,9)
##
## Available components:
##  [1] "call"           "data"           "modelName"      "n"
##  [5] "d"              "G"              "BIC"            "loglik"
##  [9] "df"             "bic"            "icl"            "hypvol"
## [13] "parameters"     "z"              "classification" "uncertainty"
## [17] "density"
```

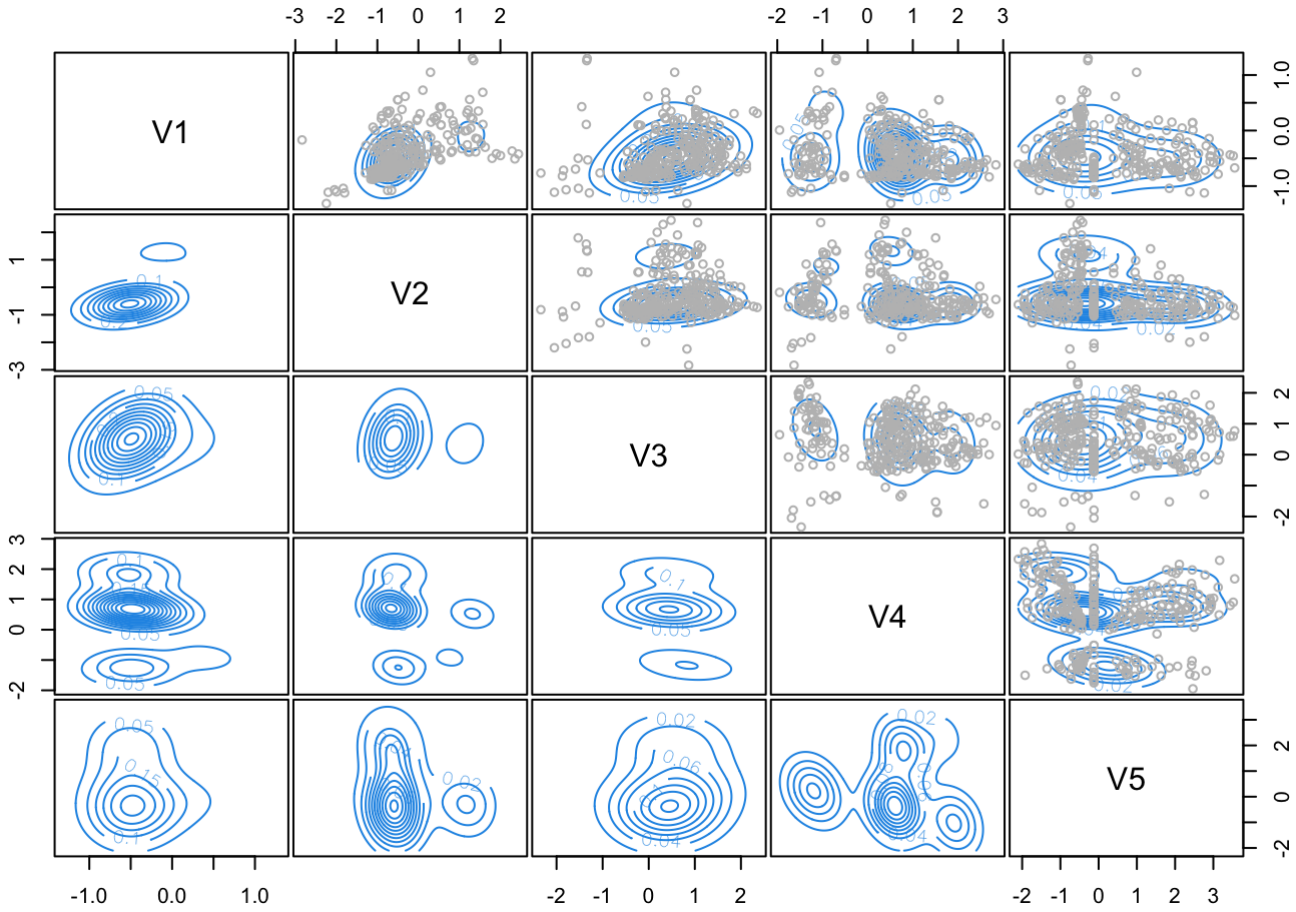
```
plot(r1e, phoneme[phoneme$Class=="Nasal",-6], what="density", col=4, points.col="gre
y")
```




```
(r2e <- densityMclust(phoneme[phoneme$Class=="Oral",-6], G=1:9, modelNames="EEE"))
```

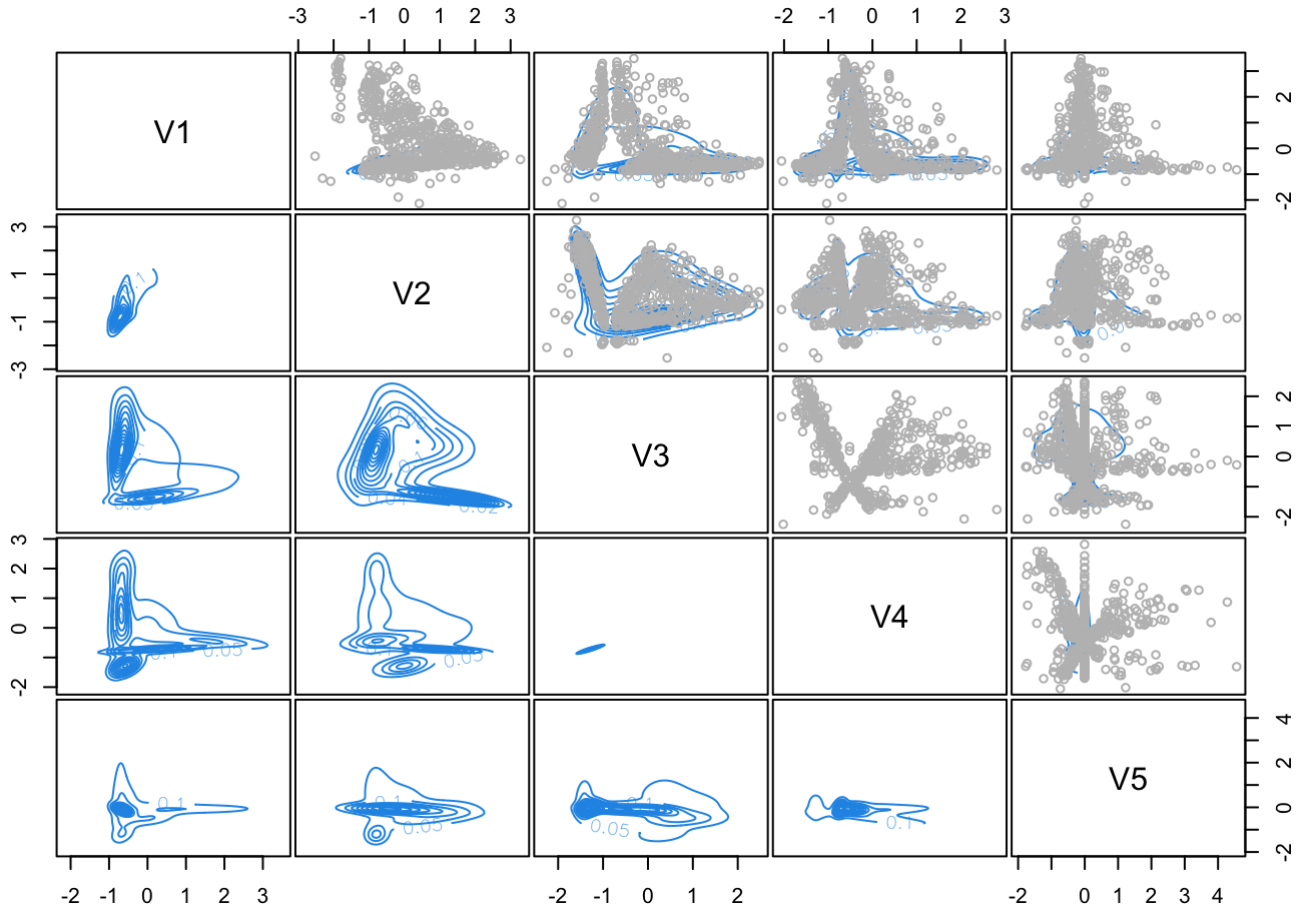
```
## 'densityMclust' model object: (EEE,8)
##
## Available components:
##  [1] "call"            "data"            "modelName"       "n"
##  [5] "d"               "G"               "BIC"             "loglik"
##  [9] "df"              "bic"             "icl"             "hypvol"
## [13] "parameters"      "z"               "classification"  "uncertainty"
## [17] "density"
```

```
plot(r2e, phoneme[phoneme$Class=="Oral",-6], what="density", col=4, points.col="grey"
)
```

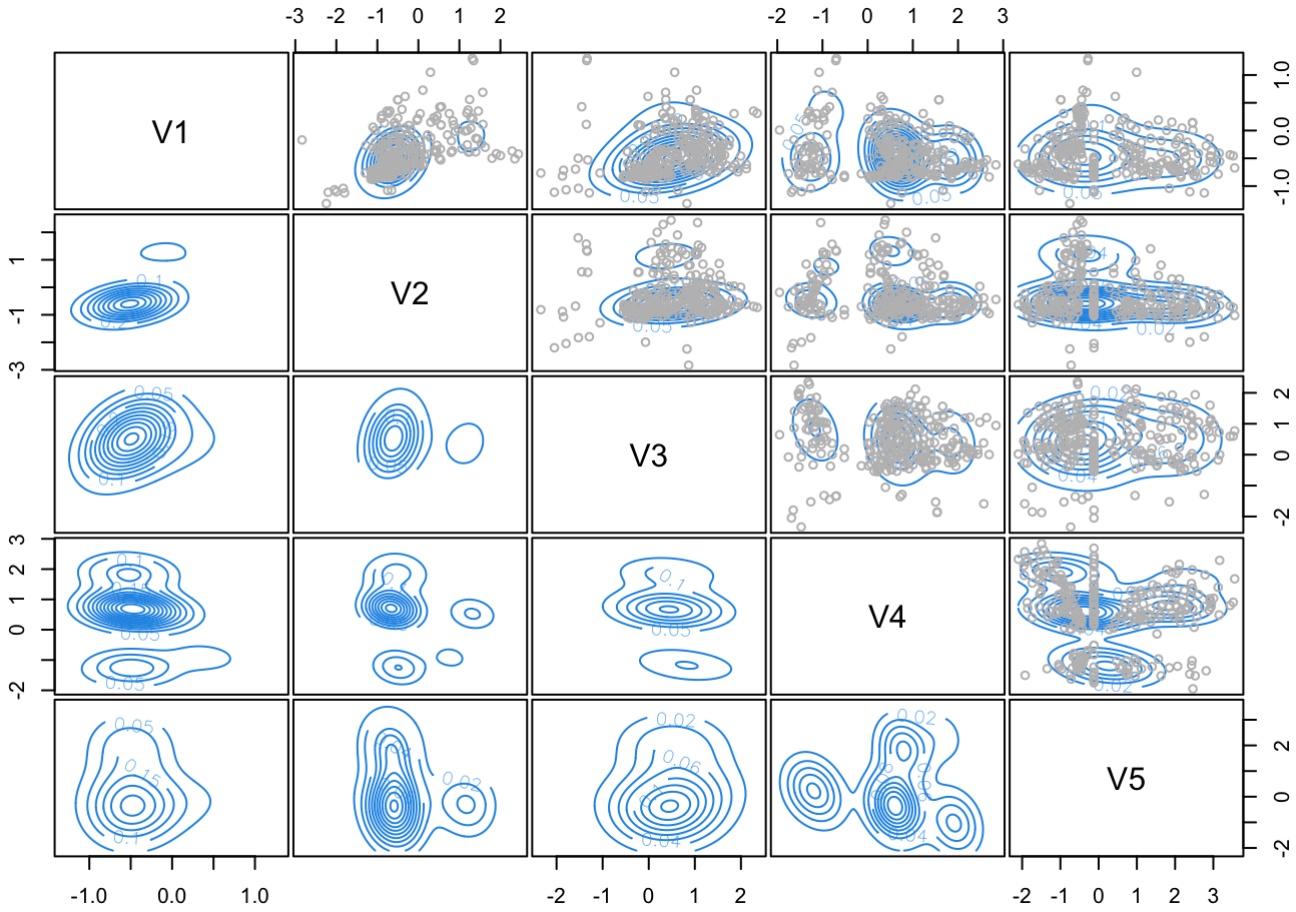


5. Repeat Task 4, but for the varying variance subfamily of normal mixtures.

```
r1v <- densityMclust(phoneme[,-6], G=1:9, modelNames="VVV")
plot(r1v, phoneme[phoneme$Class=="Nasal",-6], what="density", col=4, points.col="gre
y")
```




```
r2v <- densityMclust(phoneme[phoneme$Class=="Oral",-6], G=1:9, modelNames="VVV")
plot(r2e, phoneme[phoneme$Class=="Oral",-6], what="density", col=4, points.col="grey"
)
```



6. By applying the fundamental rule of classification, we obtain immediately a classifier from Tasks 4 and 5, respectively. For each classifier, compute its resubstitution misclassification rate.

```
p = 1/nrow(phoneme)
(mean(predict(r1e, phoneme[phoneme$Class=="Nasal", -6]) < 0.5))
```

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

```
(mean(predict(r2e, phoneme[phoneme$Class=="Oral", -6]) < 0.5))
```

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

```
(mean(predict(r1v, phoneme[phoneme$Class=="Nasal", -6]) < 0.5))
```

```
## [1] 0.9351199
```

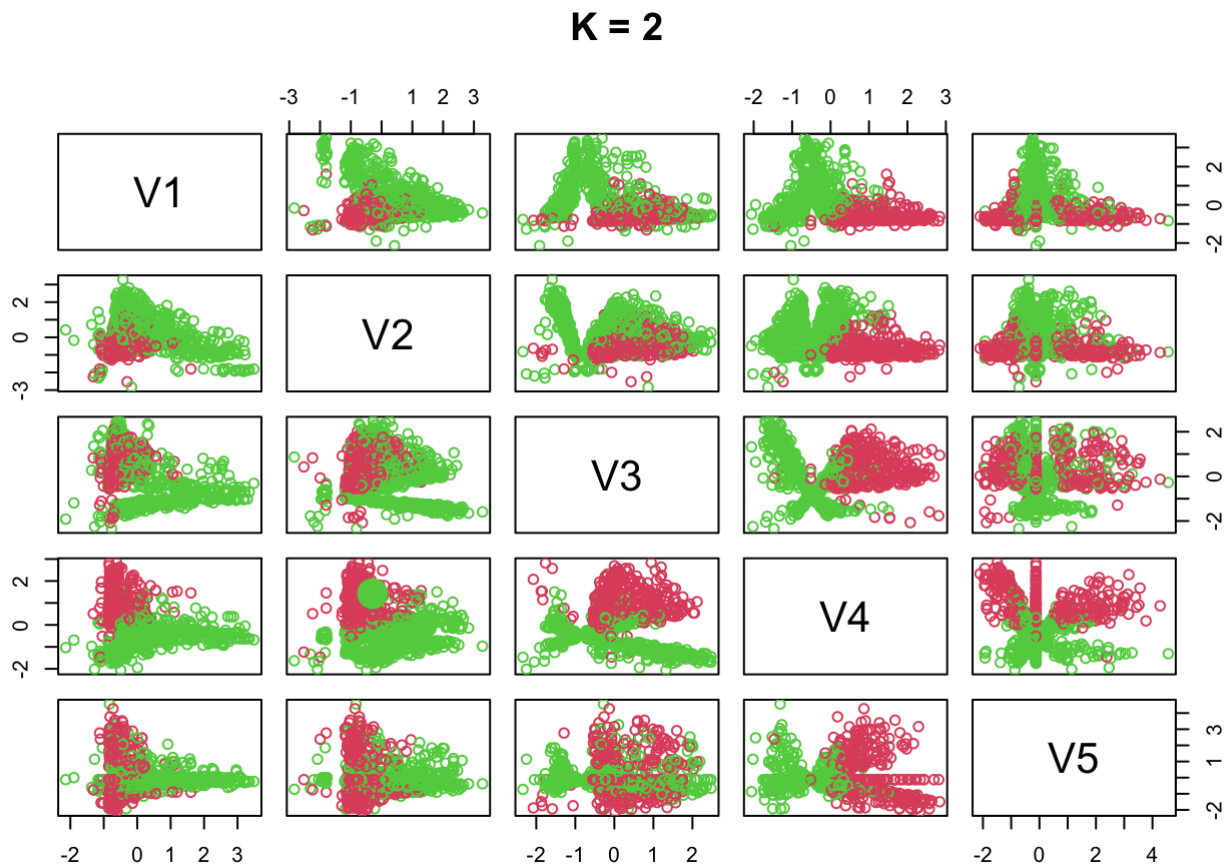
```
(mean(predict(r2v, phoneme[phoneme$Class=="Oral", -6]) < 0.5))
```

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

K-means

7. With the K-means method, find the clustering results with two clusters. Show the results in a pairwise plot of the data, using different colors and point types for observations of different clusters. (If you have successfully finished Task 1, then plot the majority cluster first.)

```
r = kmeans(phoneme[,-6], centers=2)
with(phoneme[,-6], plot(phoneme[,-6], col=r$cluster+1, main="K = 2"))
points(r$centers, col=2:3, pch=19, cex=2)
```



8. One might be curious to see if the clusters found by a clustering method are anywhere similar to the class labels that are also available in a data set (i.e., to examine the performance of an unsupervised learning method for a supervised learning problem). Compute the adjusted Rand indices for K=2,…,9 clusters as found by the K-means method, when the given class labels are contrasted. Comment on the results.

```
set.seed(769)
ari = double(5)
for(k in 2:9) {
  r = kmeans(phoneme[,-6], centers=k)
  ari[k-1] = adjustedRandIndex(phoneme$Class, r$cluster)
}
ari
```

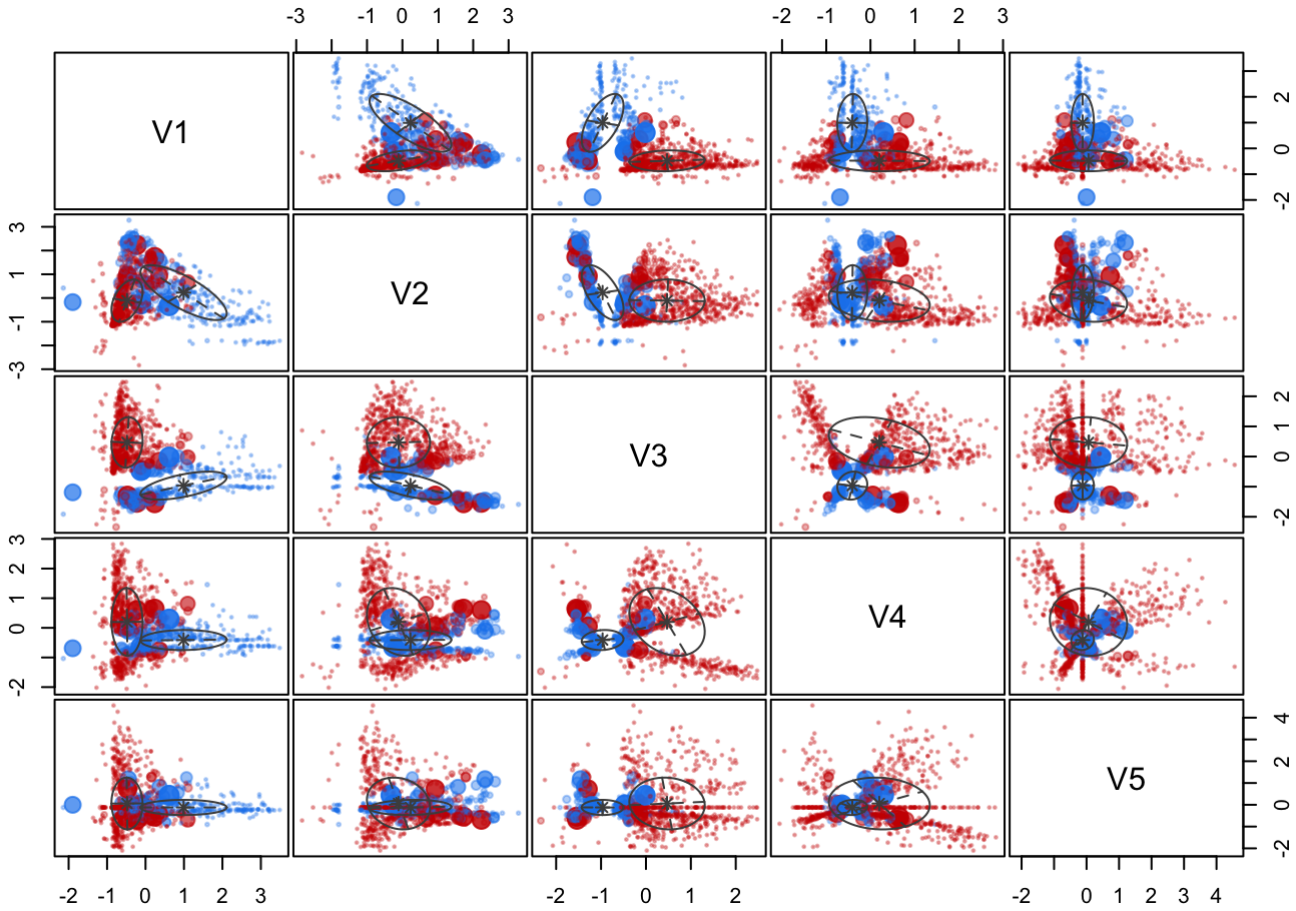
```
## [1] 0.10741874 0.11711319 0.08404257 0.09069357 0.06838202 0.07020157 0.04489206
## [8] 0.05442945
```

The clusters overlapped the classifications to a large extend. They're still quite similar even when clusters (k > 2) outnumbered the classifications, probably due to that the extra clusters just grouped the outliers.

Mixture-based clustering

9. Using the varying variance subfamily of multivariate normal mixtures, find the clustering results with two clusters, and show the results in a pairwise plot as in Task 7. Also compute the adjusted Rand indices for K=2,…,9 clusters (as done in Task 8).

```
r = Mclust(phoneme[, -6], G=2, modelNames="VVV")
plot(r, "uncertainty")
```




```
ari = double(5)
for(k in 2:9) {
  r = Mclust(phoneme[,-6], G=k, modelNames="VVV")
  ari[k-1] = adjustedRandIndex(phoneme$Class, r$classification)
}
ari
```

```
## [1] 0.02963741 0.05014537 0.07091669 0.05169660 0.03648900 0.03973215 0.03236527
## [8] 0.03569147
```

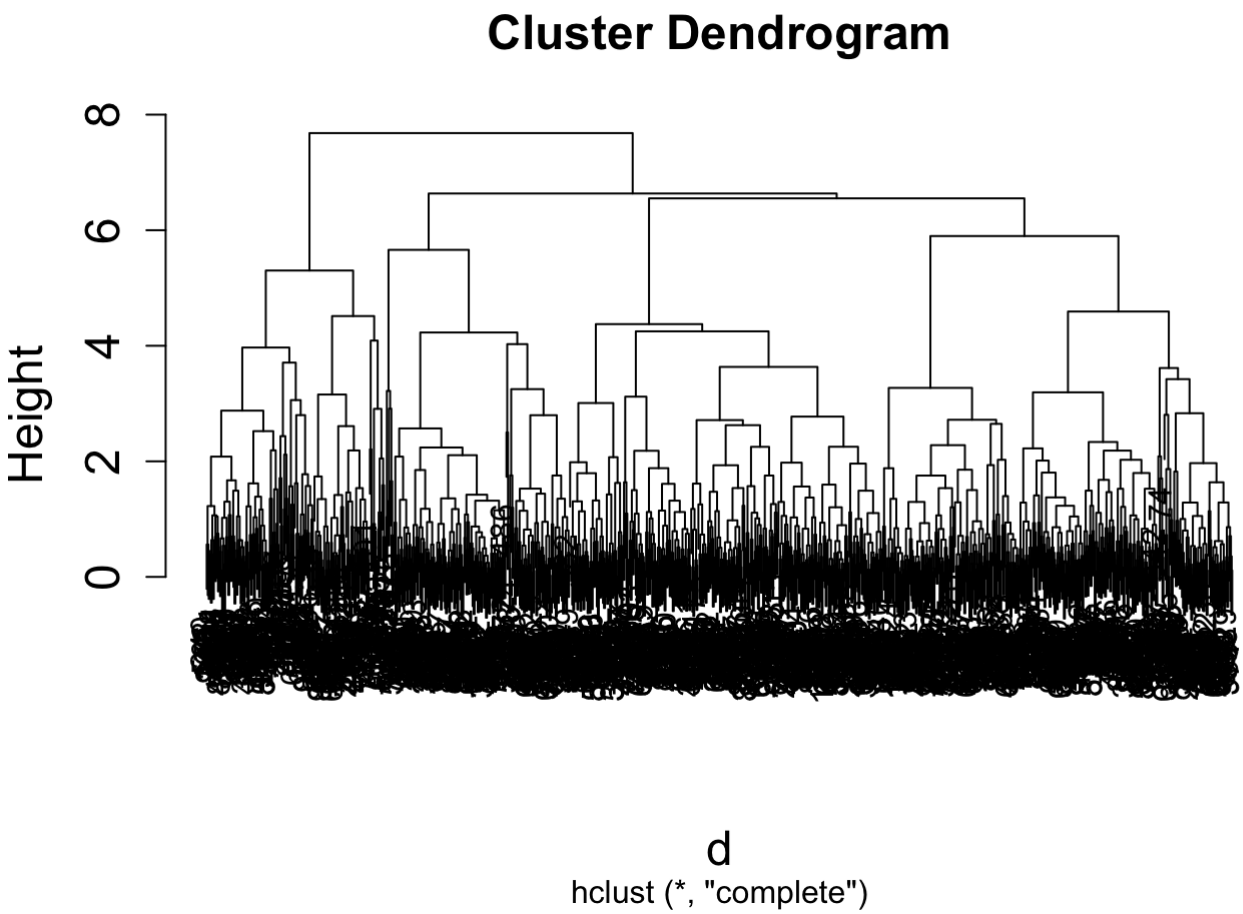
Hierarchical Clustering

10. Produce a dendrogram of the hierarchical clustering results using the complete and the single linkage method, respectively. Explain why they look very much different.

```
d = dist(phoneme[, -6])
r1 = hclust(d)
names(r1)
```

```
## [1] "merge"       "height"      "order"       "labels"      "method"
## [6] "call"        "dist.method"
```

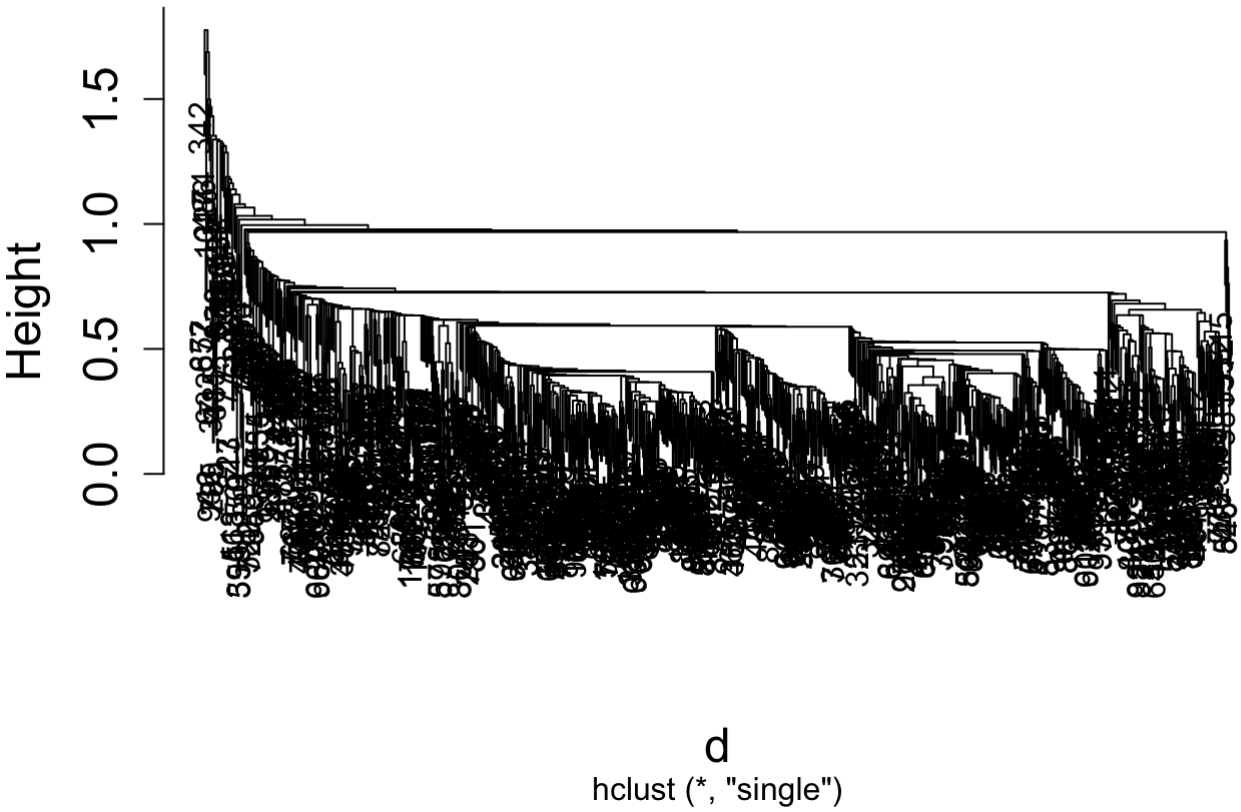
```
plot(r1, cex.axis=1.5, cex.lab=1.5, cex.main=1.5)
```




```
r2 = hclust(d, method="single")
names(r2)
```

```
## [1] "merge"       "height"      "order"       "labels"      "method"
## [6] "call"        "dist.method"
```

```
plot(r2, cex.axis=1.5, cex.lab=1.5, cex.main=1.5)
```



The complete linkage uses the largest pairwise observation dissimilarity, which tends to be compact (as we can see from the dendrogram, it merged the clusters faster. And the single linkage is the other way around.

11. Compute the adjusted Rand indices for K=2,…,9 clusters produced by the complete and the single linkage method, respectively (as done in Task 8).

```
r = hclust(d)

ari = double(5)
for(k in 2:9) {
  r = hclust(d)
  cutree(r, k)
  ari[k-1] = adjustedRandIndex(phoneme$Class, cutree(r, k))
}
ari
```

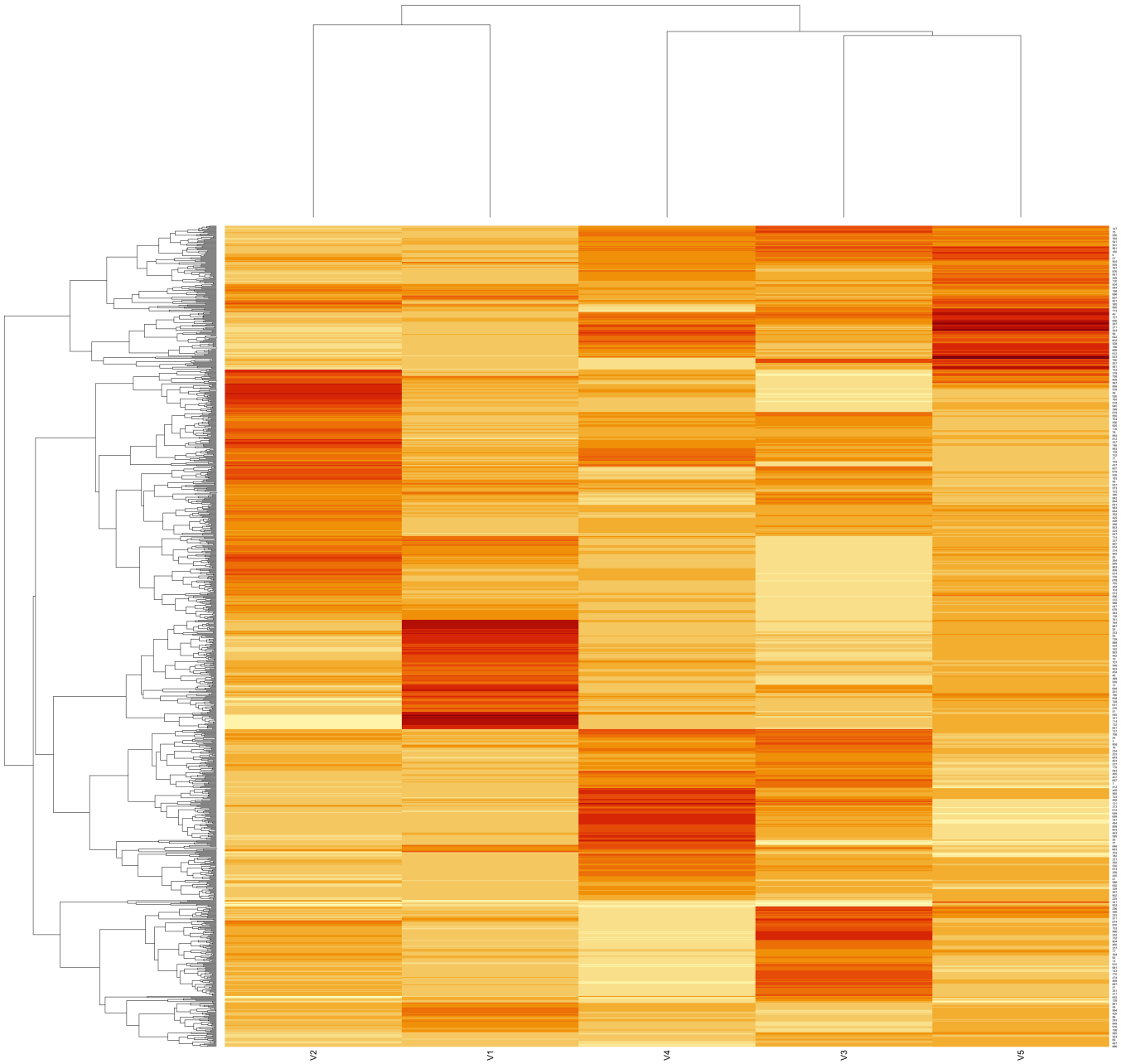
```
## [1] 0.15231471 0.08742364 0.04782010 0.06935799 0.07016402 0.07475356 0.08060478
## [8] 0.08153396
```

```
r = hclust(d)

ari = double(5)
for(k in 2:9) {
  r = hclust(d, method="single")
  cutree(r, k)
  ari[k-1] = adjustedRandIndex(phoneme$Class, cutree(r, k))
}
ari
```

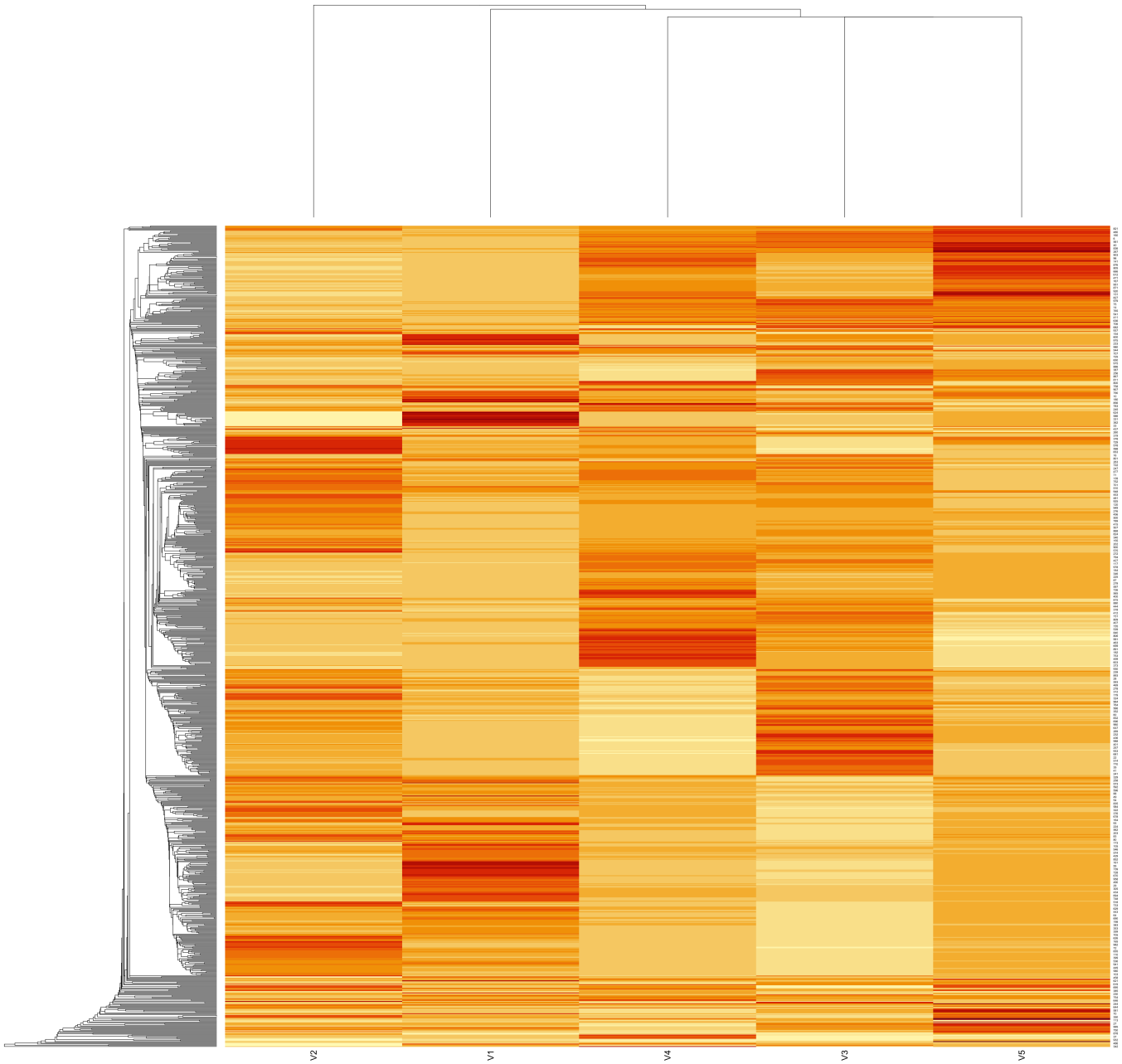
```
## [1] -0.0011781335  0.0005002740 -0.0006730085  0.0021753775  0.0050229980
## [6]  0.0038415529  0.0083236625  0.0143949554
```

12. Produce the heatmaps for both the complete and single linkage methods.

```
single <- function(d) {
  hclust(d, method="single")
}

heatmap(as.matrix(phoneme[, -6]), scale="column", distfun=dist, hclustfun=hclust,
        margins=c(15,5))
```




```
heatmap(as.matrix(phoneme[, -6]), scale="column", distfun=dist, hclustfun=single,
        margins=c(15,5))
```



Summary

In this lab, we explored the data set phoneme which has 5 numeric predictors and one response variable class: Nasal (nasal vowels) and Oral (oral vowels).

First we tried to estimate the density to approximate the classification labels, the accuracy wasn't so good. Then we moved to clusters methods, it can predict better as an alternative to the traditional supervise learning methods (this time we use rand indicies to gauge the prediction.) Lastly, we compared the complete linkage and the single linkage methods in the hierarchical clustering, and observed the differences by looking into denodrogram and heatmap to see how the trees were constructed in each method.