Hierarchical Clustering of Gene Data

Anthony Lighterness

03/04/2020

R Setup of Working Directory

R Packages

```
#install.packages("wordspace")
#install.packages("dbscan")
# uncomment if system needs packages to be installed
```

R Libraries

```
library(foreign)
library(tidyverse)
library(wordspace)
library(dplyr)
library(dbscan)
library(ggplot2)
library(reshape2)
```

Set Figure Margins

```
par(mar=c(1, 1, 1, 1))
```

Activity 1: Clustering Cancerous Tissue Samples

Question 1 read and import the data.

```
leuk_data = read.arff("golub-1999-v1_database.arff")
```

Question 2 set aside rightmost column, Classe.

```
leuk_classe = leuk_data %>%
  select(Classe) %>%
  as.matrix()

leuk_data_int = leuk_data %>%
  select(-Classe)    # Remove Classe variable
```

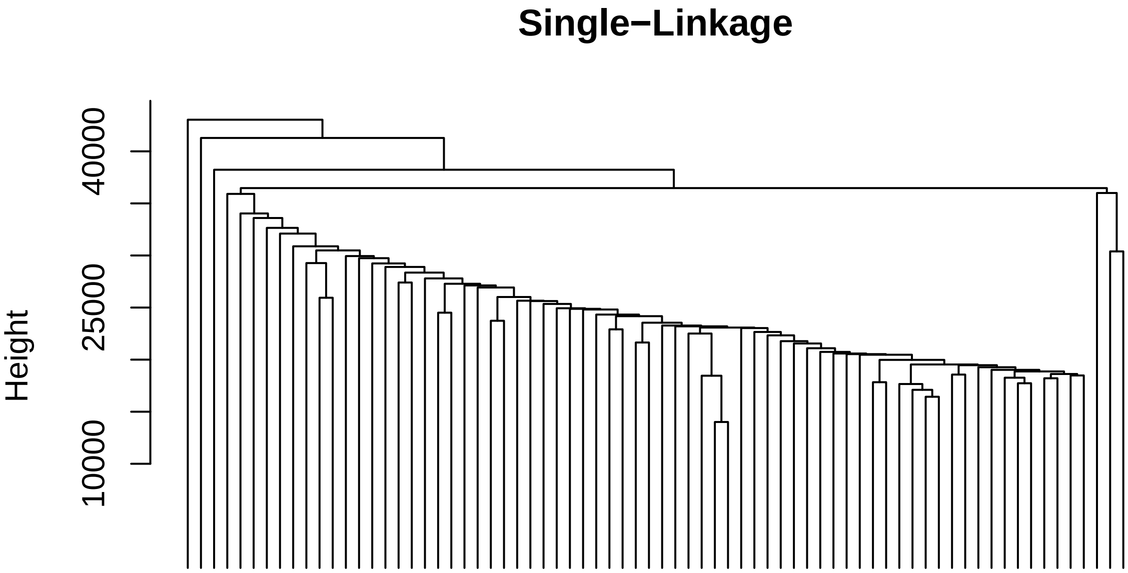
Question 3 compute pairwise euclidean distances by each row.

```
leuk_dist = dist(leuk_data_int, method="euclidean")
```

Question 4 single-linkage clustering algorithm with dendrogram.

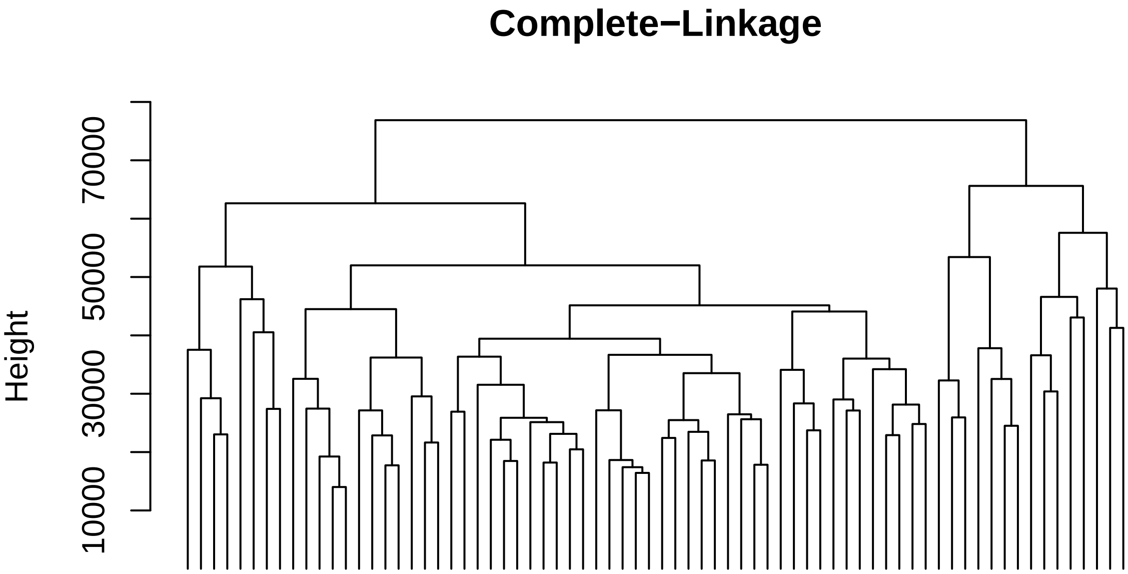
```
SL_leuk = hclust(leuk_dist, method="single")
```

```
# Plot Single-Linkage Dendrogram
plot(SL_leuk, main="Single-Linkage", xlab="", sub="", hang=-1, labels=FALSE)
```



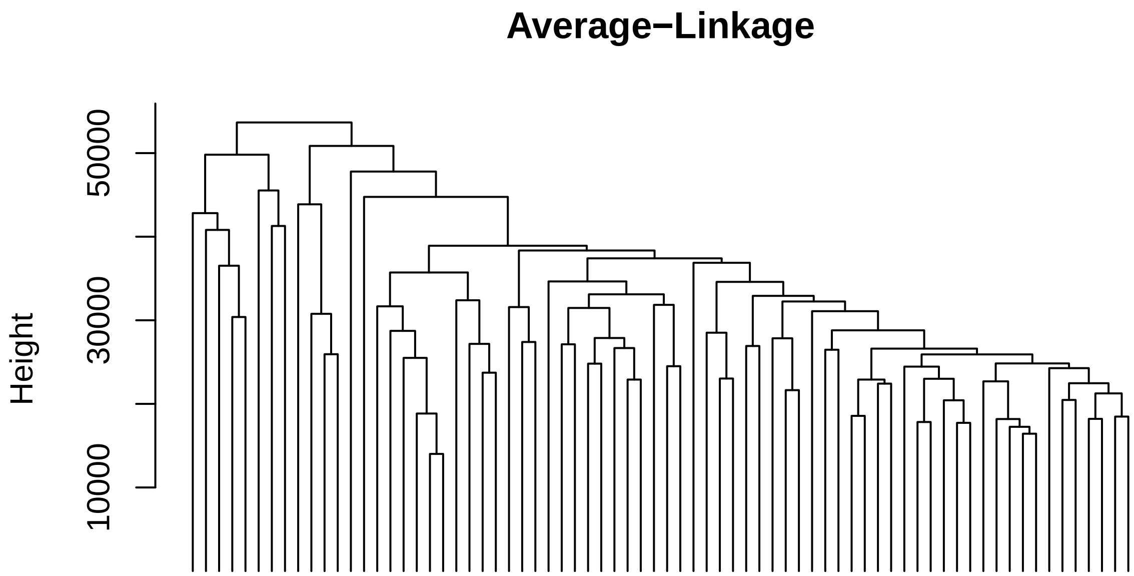
Question 5 complete-linkage clustering algorithm with dendrogram.

```
CL_leuk = hclust(leuk_dist, method = "complete")

# Plot Complete-Linkage Dendrogram
plot(CL_leuk, main="Complete-Linkage", xlab="", sub="", hang=-1, labels=FALSE)
```



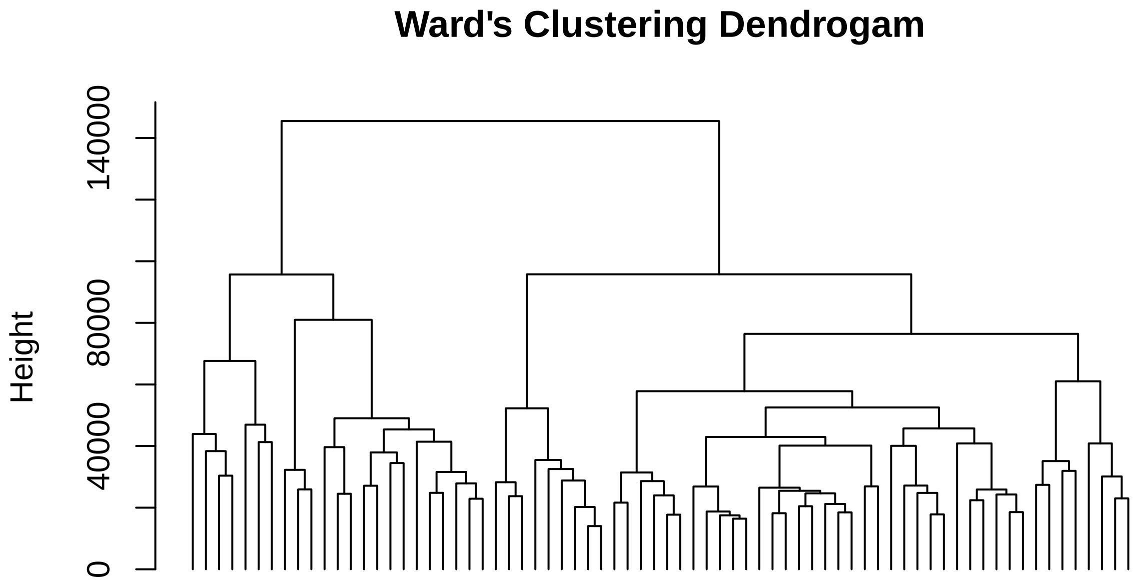
Question 6 average-linkage clustering algorithm with dendrogram.

```
AL_leuk = hclust(leuk_dist, method = "average")

# Plot Average-Linkage Dendrogram
plot(AL_leuk, main="Average-Linkage", xlab="", sub="", hang=-1, labels=FALSE)
```



Question 7 ward's clustering algorithm with dendrogram.

```r
wards_leuk = hclust(leuk_dist, method = "ward.D2")

# Plot Ward's Dendrogram
plot(wards_leuk, main="Ward's Clustering Dendrogam", xlab="", sub="", hang=-1, labels=FALSE)
```



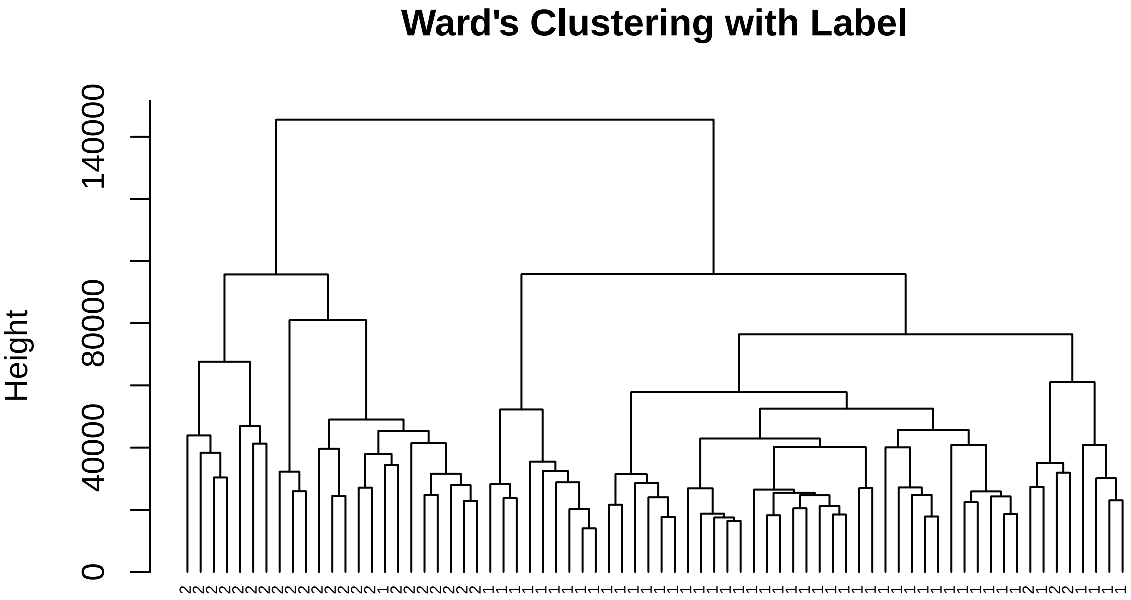
Question 8 compare the dendrograms - which generates clearer clusters?

Figure 1 shows the dendrograms of Single-Linkage (SL), Average-Linkage (AL), Complete-Linkage (CL), and Ward's clustering algorithms. While the former two algorithms yield extended clusters to which single single leaves are fused one by one, the latter two produce more evenly sized clusters. Furthermore, it is apparent that the CL and Ward's clustering algorithms produce the clearest distinctions between specific groups of clusters, which is indicated by the vertical lengths of branches separating fusion points between clusters. Meanwhile, the SL and AL dendrograms show a greater number of clusters that are nearly indistinguishable from one another due to much shorter vertical distances. When looking at the top-most fusion, both the SL and AL dendrograms show much shorter distances between subsequent, lower clusters. In contrast, the

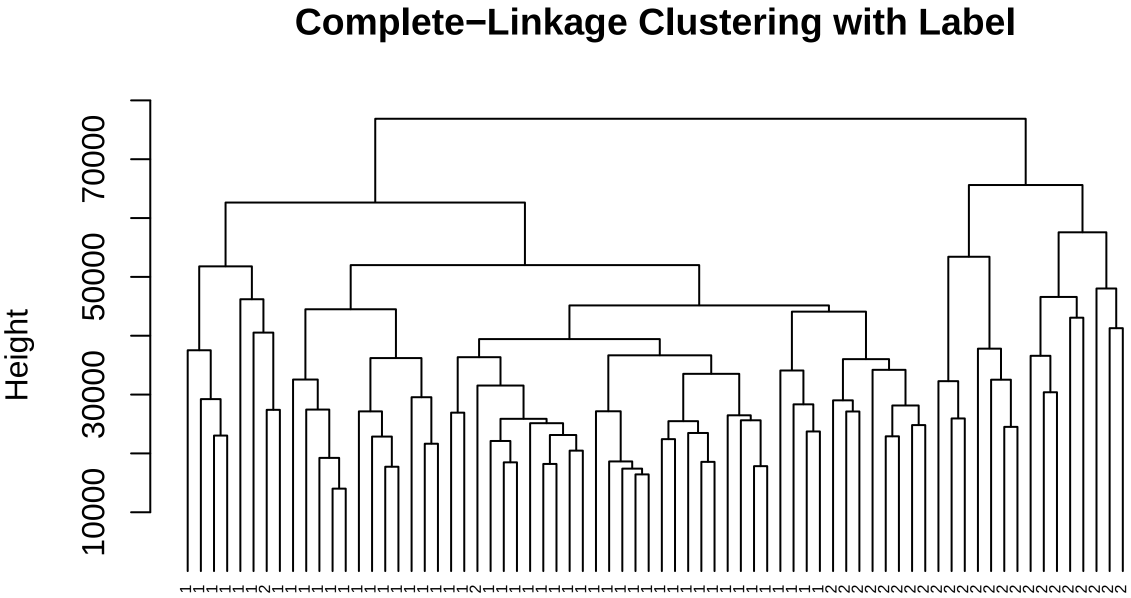
Ward's dendrogram shows a distance of approximately 40,000 between the top-most fusion and the two main clusters it fuses, or splits into.This distance in the CL dendrogram is approximately 10,000-12,000 units, which again is much larger than the mere 2,000-5,000 length distance observed in the SL and AL algorithm dendrograms. Overall, these observations makes the CL and Ward's dendrogram more balanced and favourable to distinguish between different major clusters.

Question 9 use class labels for previously plotted dendrograms.

```r
plot(wards_leuk, main="Ward's Clustering with Label",
     xlab="", sub="", hang=-1, labels=leuk_classe, cex = 0.5)
```



```r
plot(CL_leuk, main="Complete-Linkage Clustering with Label",
     xlab="", sub="", hang=-1, labels=leuk_classe, cex = 0.5)
```



The class label does in fact show prominent clusters in both Ward's and CL clustering dendrograms. This means that the two subtypes of leukaemia have been clearly separated.

Question 10 z-score normalisation.

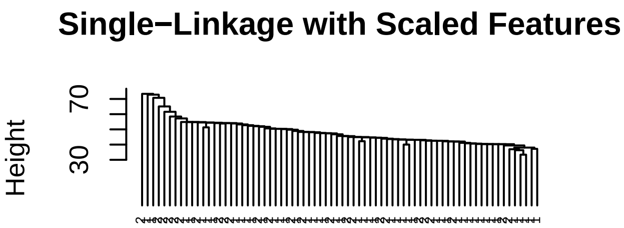
```
# Take hclust_matrix, which is the original dataframe without the classe variable
leuk_data_scaled = leuk_data_int %>%
  scale # apply scalling to each column of the matrix (genes)

# Check that we get mean of 0 and sd of 1
summary(apply(leuk_data_scaled, 2, mean)) # mean for each column = ~0
```

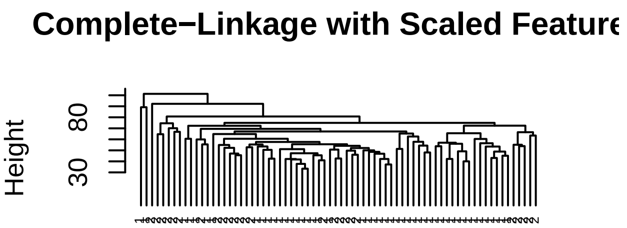
```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -1.308e-16 -2.426e-17  7.815e-19  1.495e-18  2.778e-17  1.216e-16
```

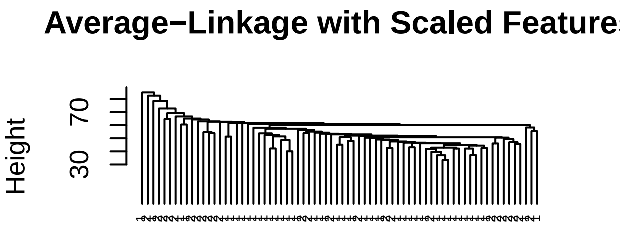
```
summary(apply(leuk_data_scaled, 2, sd)) # each column sd = 1
```

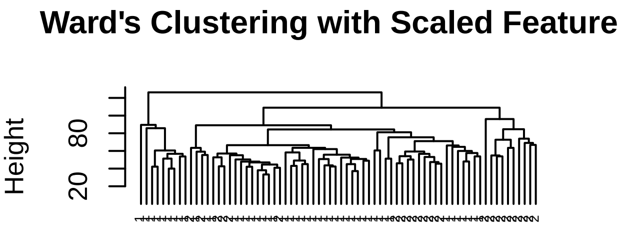
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       1       1       1
```

```
# Plot scaled hierarchical clustering algorithms together
par(mfrow=c(2,2))
# Single-Linkage
plot(hclust(dist(leuk_data_scaled), method="single"),
     main="Single-Linkage with Scaled Features", xlab="", sub="",
     labels=leuk_classe, hang=-1, cex=0.5)

# Complete-Linkage
plot(hclust(dist(leuk_data_scaled), method="complete"),
     main="Complete-Linkage with Scaled Features", xlab="", sub="",
     labels=leuk_classe, hang=-1, cex=0.5)

# Average-Linkage
plot(hclust(dist(leuk_data_scaled), method="average"),
     main="Average-Linkage with Scaled Features", xlab="", sub="",
     labels=leuk_classe, hang=-1, cex=0.5)

# Ward's Clustering
plot(hclust(dist(leuk_data_scaled), method="ward.D2"),
     main="Ward's Clustering with Scaled Features", xlab="", sub="",
     labels=leuk_classe, hang=-1, cex=0.5)
```









All four clutering dendrograms suffer from worsened results due to the normalisation procedure. This is due to excessive bridging and unclear distinction between clusters.

Activity 2: Clustering Genes Part A

Question 11 read and import yeast data.

```
yeast_data = read.arff("yeast.arff")
```

Question 12 set aside classe variable.

```
yeast_labels = as.matrix(yeast_data$Classe)

# Remove Classe variable
yeast_matrix = yeast_data %>%
  select(-Classe) %>%
  as.matrix()
```

Question 13 pearson similarity 205x205 matrix.

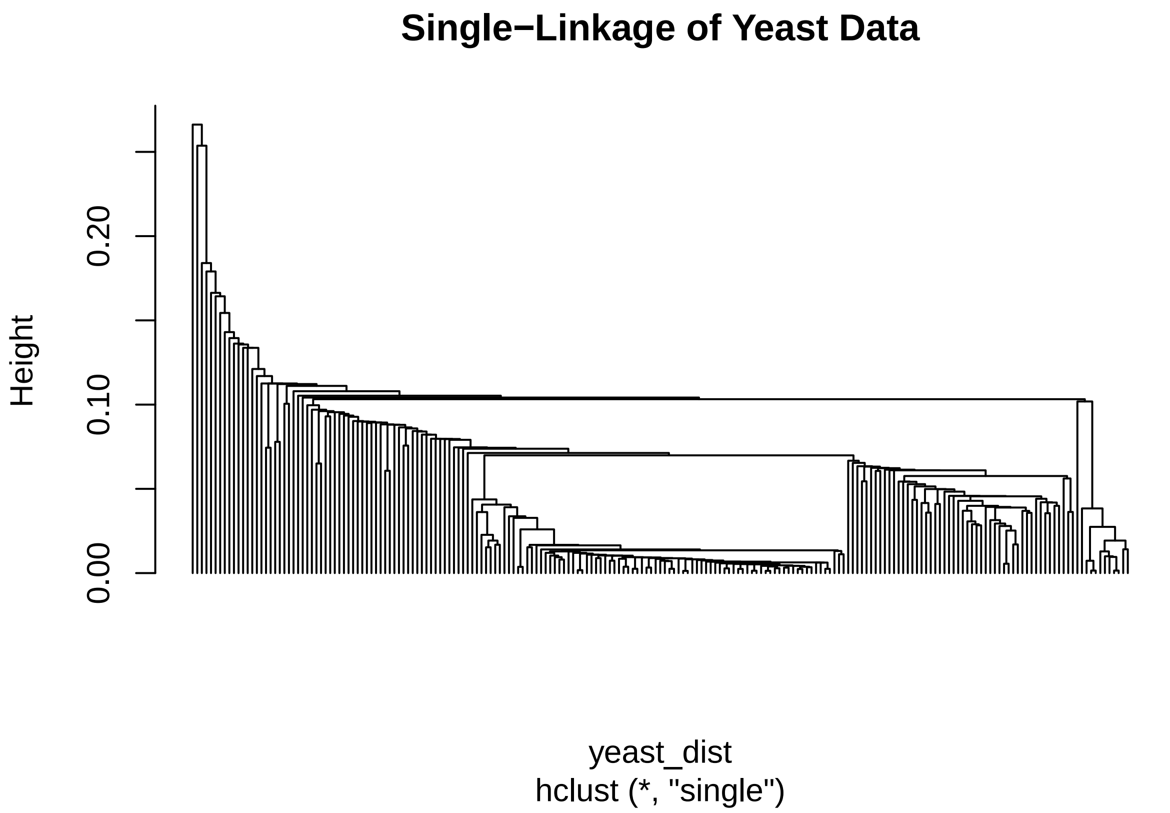
```
yeast_cor = yeast_matrix %>%
  t() %>% # transpose the matrix
  cor(method="pearson") # Pearson correlation/similarity matrix

# Convert to dissimilarity = 1-similarity and coerce into dist type
yeast_dist = as.dist(1 - abs(yeast_cor))/2

summary(yeast_dist) # summary shows values range from 0 to +1
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001316 0.156897 0.262309 0.256937 0.364078 0.499975
```

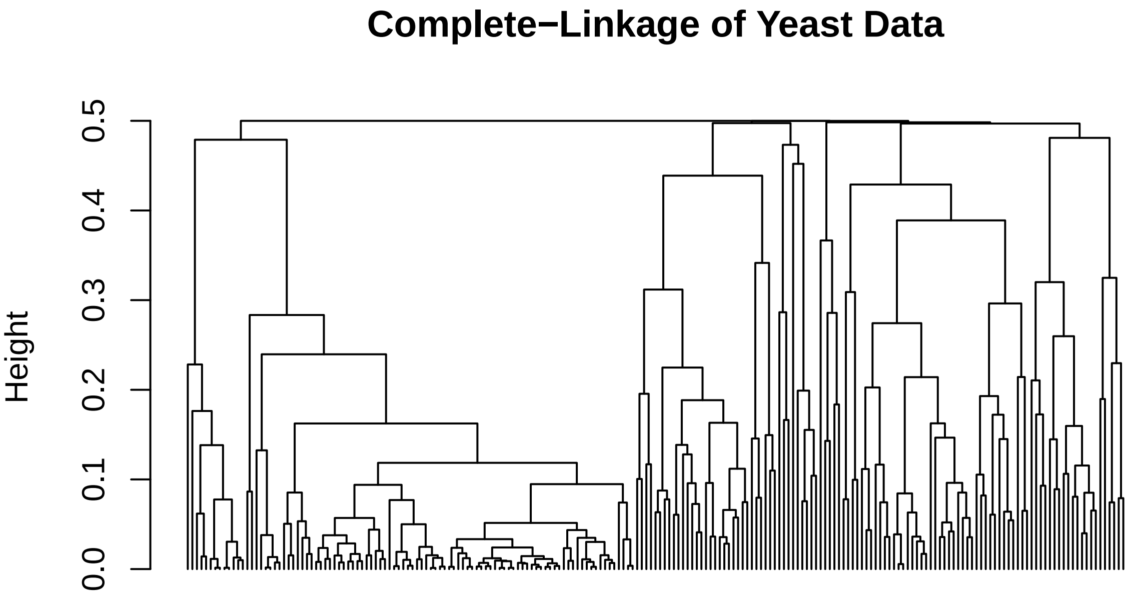
Summary of distance matrix, yeast_dist, confirms successful conversion of pearson similarity matrix.

Question 14 repeat items 4-9 from activity 1.

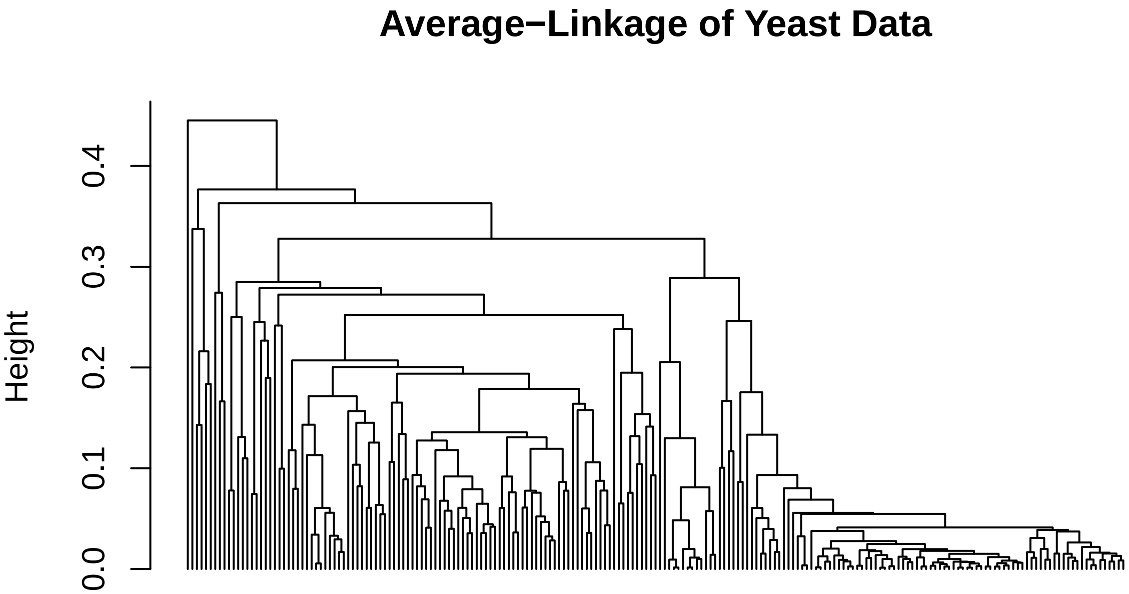
```
# Plot hierarchical clustering dendrograms together using dissimilarity matrix yeast_dist
# Single-Linkage
yeast_SL = hclust(yeast_dist, method="single")
plot(yeast_SL, main="Single-Linkage of Yeast Data", labels=FALSE, hang=-1)
```



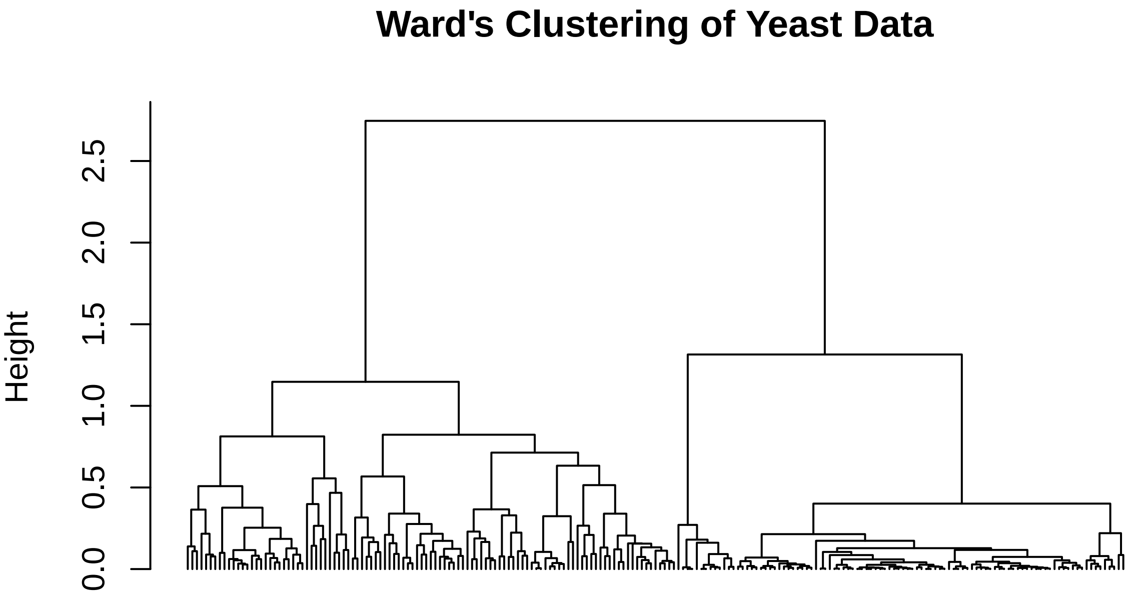
```
# Complete-Linkage
yeast_CL = hclust(yeast_dist, method="complete")
plot(yeast_CL, main="Complete-Linkage of Yeast Data",
     xlab="", sub="", labels=FALSE, hang=-1)
```



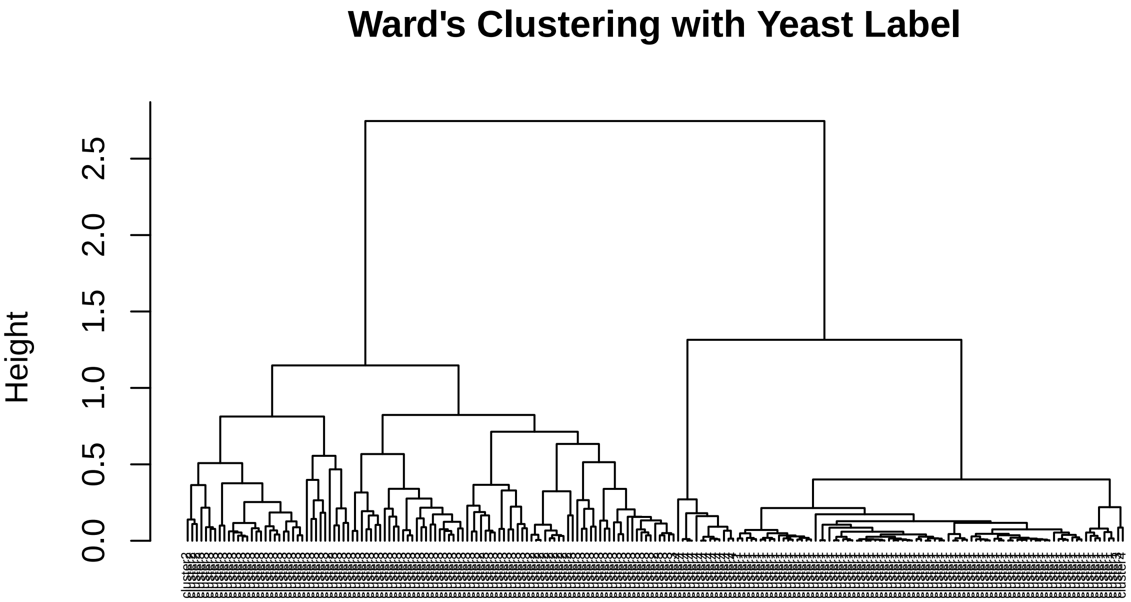
```
# Average-Linkage
yeast_AL = hclust(yeast_dist, method="average")
plot(yeast_AL, main="Average-Linkage of Yeast Data",
     xlab="", sub="", labels=FALSE, hang=-1)
```



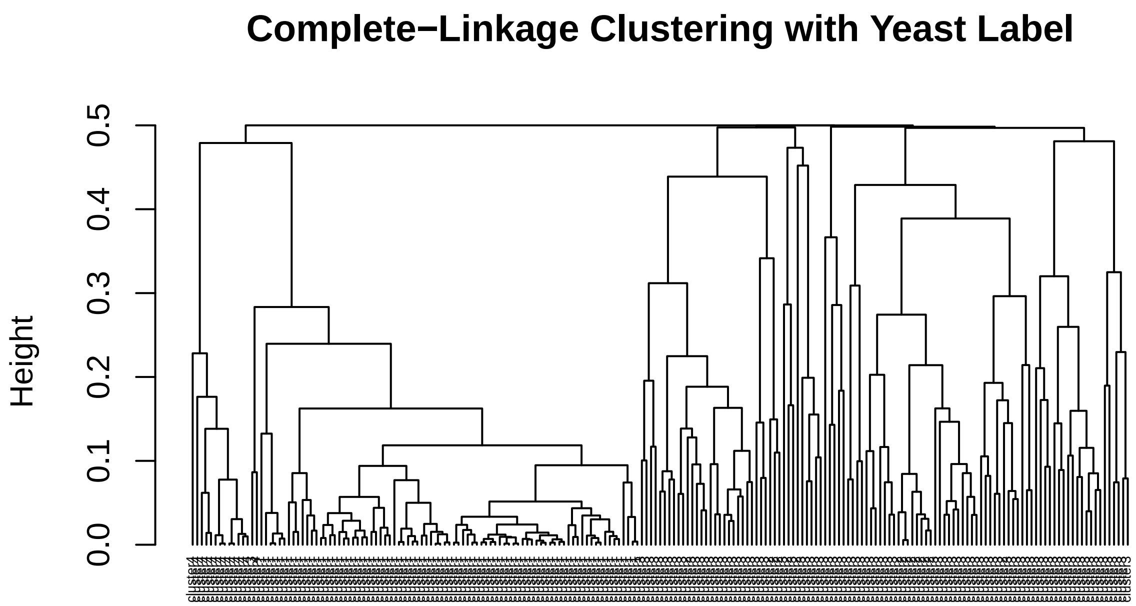
```
# Ward's Clustering
yeast_Ward = hclust(yeast_dist, method="ward.D2")
plot(yeast_Ward, main="Ward's Clustering of Yeast Data",
     xlab="", sub="", labels=FALSE, hang=-1)
```



```r
# Plot Ward's and Complete-Linkage dendrograms with Yeast labels
plot(yeast_Ward, main="Ward's Clustering with Yeast Label",
     xlab="", sub="", hang=-1, labels=yeast_labels, cex=0.4)
```



```r
# CL Dendrogram with class label:
plot(yeast_CL, main="Complete-Linkage Clustering with Yeast Label",
     xlab="", sub="", hang=-1, labels=yeast_labels, cex=0.4)
```



Activity 3: Clustering Genes Part B

Question 15 rescale yeast_matrix in a row-wise fashion so that each row has magnitude 1, i.e. euclidean row-wise normalisation.

Euclidean normalision given by |x| = sqrt(sum(i) (x_i)^2)

```
yeast_normalise = yeast_matrix %>%
  normalize.rows(method="euclidean")
```

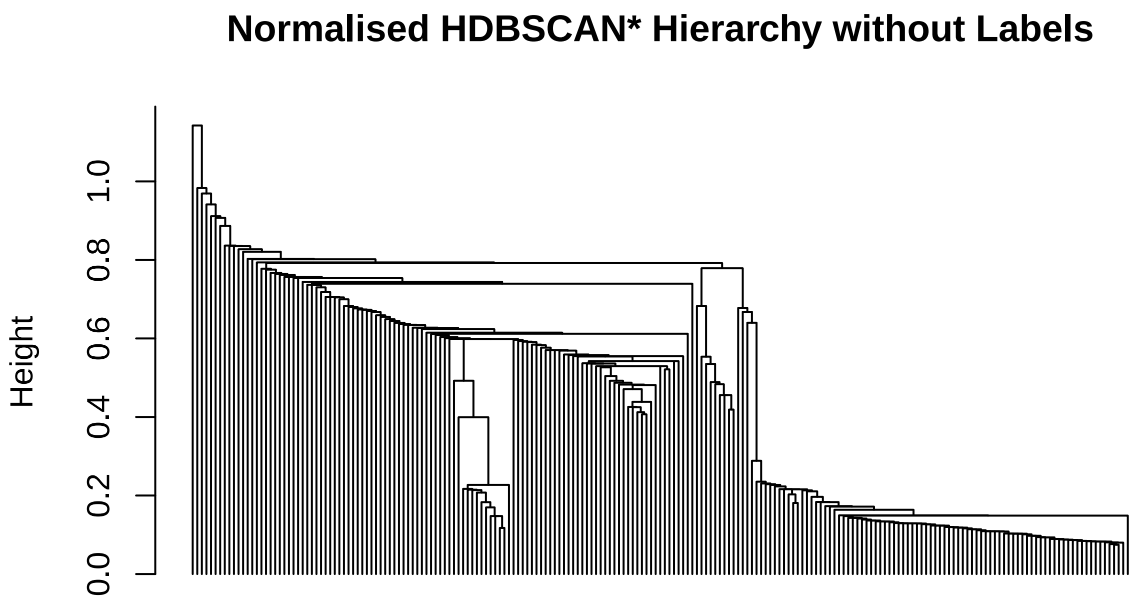
sqrt(sum(dat_Rescaled[i,]^2)) == 1 for any random observation, confirms that yeast_normalise is correctly rescaled

Question 16 HDBSCAN* with euclidean distance at MinPts = 5.

```
set.seed(0)

# HDBSCAN* for yeast_normalised
hdbs_yeast_norm = hdbscan(yeast_normalise, minPts = 5)
hdbs_yeast_norm
```

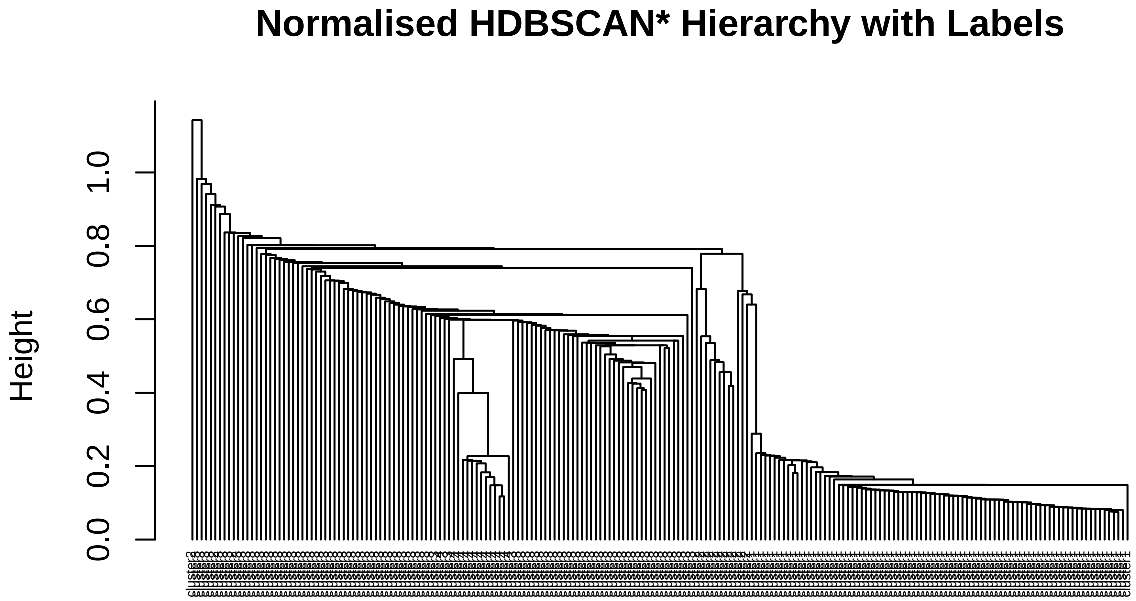
```
## HDBSCAN clustering for 205 objects.
## Parameters: minPts = 5
## The clustering contains 4 cluster(s) and 59 noise points.
##
##  0  1  2  3  4
## 59 13 38  9 86
##
## Available fields: cluster, minPts, cluster_scores,
##                   membership_prob, outlier_scores, hc
```

```
# Plot DBSCAN* dendrograms
# Without classe label
plot(hdbs_yeast_norm$hc, main="Normalised HDBSCAN* Hierarchy without Labels",
     xlab="", sub="", hang=-1, labels=FALSE)
```



Normalised HDBSCAN* Hierarchy without Labels

```r
# With classe label
plot(hdbs_yeast_norm$hc, main="Normalised HDBSCAN* Hierarchy with Labels",
     xlab="", sub="", hang=-1, labels=yeast_labels, cex=0.4)
```



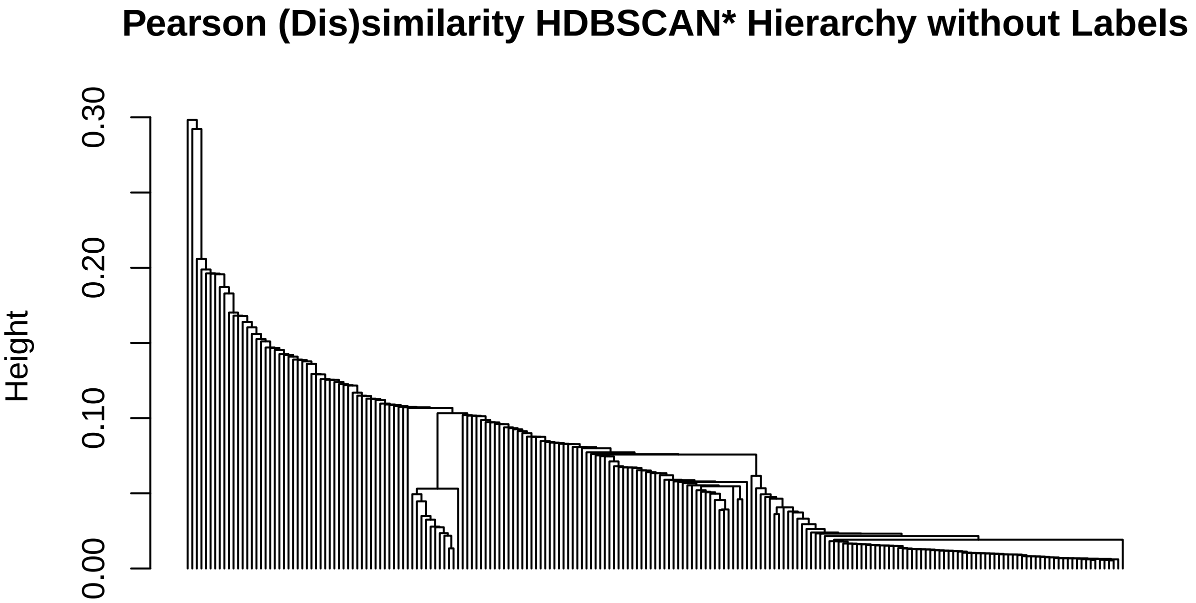
Normalised HDBSCAN* Hierarchy with Labels

```r
# Repeat as in Item 14, using dissimilarity matrix as calculated in item 13
# yeast_dist is a pearson dissimilarity calculated matrix
hdbs_yeast_pearson = hdbscan(yeast_dist, minPts = 5)
hdbs_yeast_pearson
```

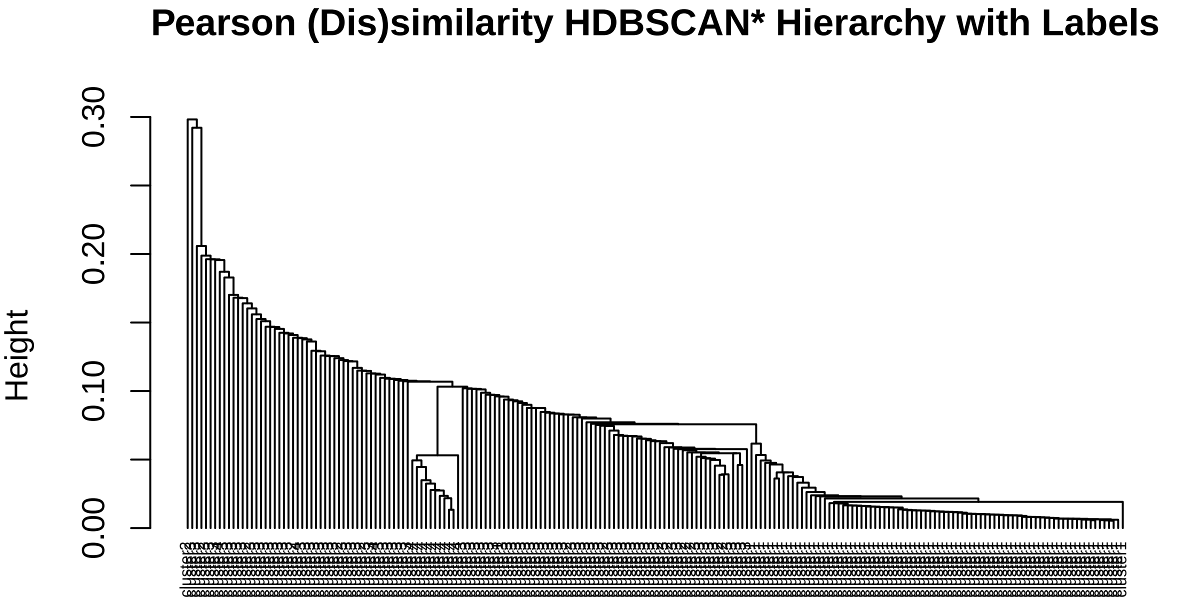
```
## HDBSCAN clustering for 205 objects.
## Parameters: minPts = 5
## The clustering contains 3 cluster(s) and 78 noise points.
##
##  0  1  2  3
## 78 11 34 82
##
```

```
## Available fields: cluster, minPts, cluster_scores,
##                   membership_prob, outlier_scores, hc
```

```
# Plot without labels
plot(hdbs_yeast_pearson$hc, main="Pearson (Dis)similarity HDBSCAN* Hierarchy without Labels",
     xlab="", sub="", hang=-1, labels=FALSE)
```


Pearson (Dis)similarity HDBSCAN* Hierarchy without Labels

```
# With classe label
plot(hdbs_yeast_pearson$hc, main="Pearson (Dis)similarity HDBSCAN* Hierarchy with Labels",
     xlab="", sub="", hang=-1, labels=yeast_labels, cex=0.5)
```


Pearson (Dis)similarity HDBSCAN* Hierarchy with Labels

Question 17 contingency tables

```
yeast_contingency_norm = table(yeast_labels, hdbs_yeast_norm$cluster)
yeast_contingency_norm
```

```
##
## yeast_labels  0  1  2  3  4
##     cluster1  0  0  0  0 83
##     cluster2  3  0  0  9  3
##     cluster3 55  0 38  0  0
##     cluster4  1 13  0  0  0
```

```
# Contingency Table for Pearson (Dis)similarity matrix
yeast_contingency_pearson = table(yeast_labels, hdbs_yeast_pearson$cluster)
yeast_contingency_pearson
```

```
##
## yeast_labels  0  1  2  3
##     cluster1  1  0  0 82
##     cluster2  7  0  8  0
##     cluster3 67  0 26  0
##     cluster4  3 11  0  0
```

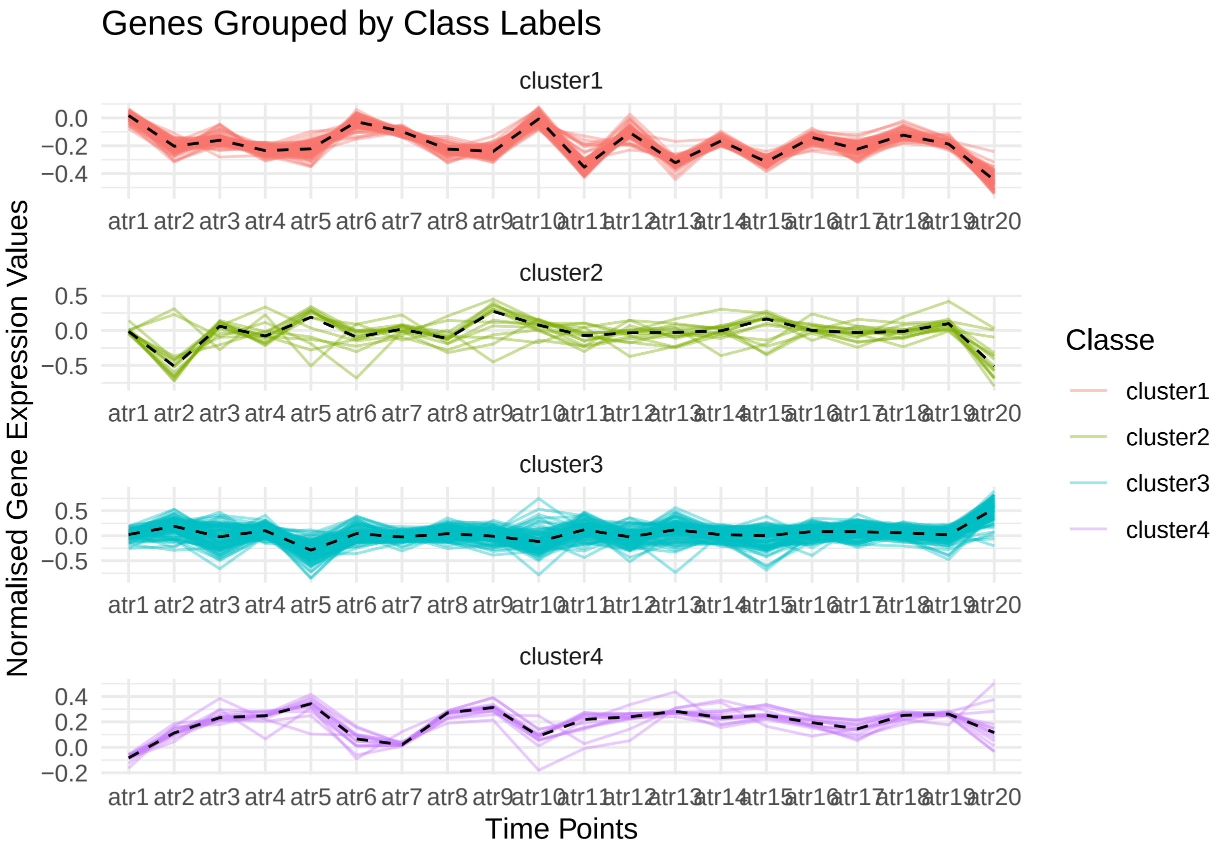
Question 18 Interpret the contingency table(s).

a. *What is the best correspondence between the four clusters and the ground truth?* The best correspondence between the four labelled clusters and the ground truth is observed between label '4' and cluster1, which is the only cluster to have all observations labelled in a single cluster. The other three clusters have some observations labelled in more than one cluster.

b. *What is the functional category for which most genes have been labelled as noise?* The most number of observations labelled as noise come from cluster3, where 55 genes are outliers identified by the HDBSCAN algorithm.

Question 19 plot genes grouped by their class labels.

```
yeast_norm2 = yeast_normalise %>%
  as.data.frame %>%
  mutate(Classe = yeast_labels) %>%
  cbind(row_no = seq(1, nrow(yeast_normalise)))

yeast_norm_melt = melt(yeast_norm2, id.vars = c("row_no", "Classe")) # setup for plot

ggplot(yeast_norm_melt, aes(x = variable, y = value, group = row_no, colour = Classe)) +
  geom_line(alpha = 0.4) +
  stat_summary(fun = median, group = 3, color = 'black', geom ='line', lty = 2) +
  ylab("Normalised Gene Expression Values") +
  xlab("Time Points") +
  ggtitle("Genes Grouped by Class Labels") +
  theme_minimal() +
  facet_wrap(~Classe, ncol=1, scale="free")
```

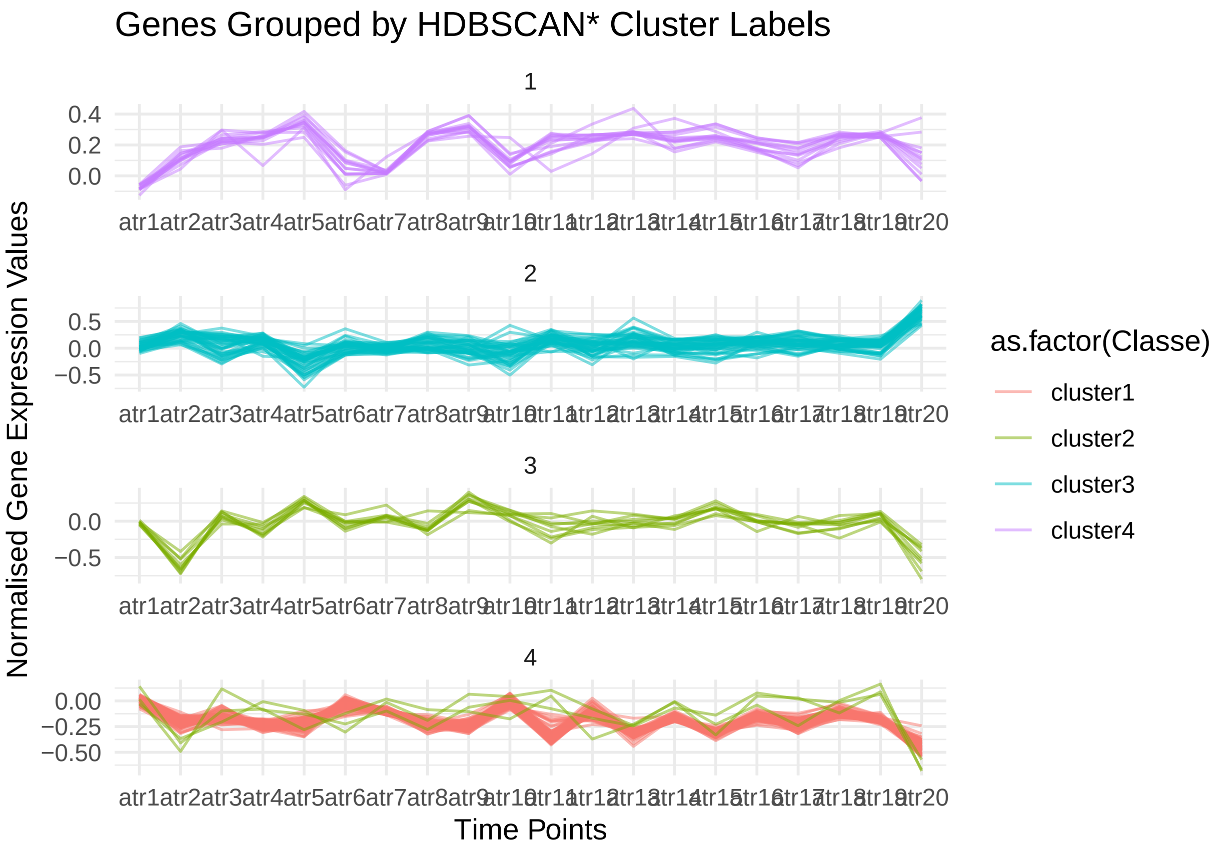


Question 20 plot genes by the HDBSCAN labelled clusters

```
yeast_norm_melt2 = yeast_norm_melt %>%
  cbind('hdbs_cluster' = hdbs_yeast_norm$cluster) %>%
  filter(hdbs_cluster !=0) # remove observations labelled as noise, i.e. '0'
summary(yeast_norm_melt2) # shows HDBS cluster min = 1
```

```
##      row_no          Classe             variable          value
##  Min.   :  1.00   Length:2920        atr1    : 146   Min.   :-0.79754
##  1st Qu.: 37.00   Class :character   atr2    : 146   1st Qu.:-0.21588
##  Median : 73.50   Mode  :character   atr3    : 146   Median :-0.10426
##  Mean   : 87.33                      atr4    : 146   Mean   :-0.07419
##  3rd Qu.:140.00                      atr5    : 146   3rd Qu.: 0.05125
##  Max.   :205.00                      atr6    : 146   Max.   : 0.89371
##                                      (Other):2044
##   hdbs_cluster
##  Min.   :1.000
##  1st Qu.:2.000
##  Median :4.000
##  Mean   :3.151
##  3rd Qu.:4.000
##  Max.   :4.000
##
```

```
# plot with HDBSCAN* cluster classifications
ggplot(yeast_norm_melt2, aes(x = variable, y = value, group = row_no,
                             colour = as.factor(Classe)), fill = as.factor(x)) +
  geom_line(stat="identity", alpha = 0.5) +
  ylab("Normalised Gene Expression Values") +
  xlab("Time Points") +
  ggtitle("Genes Grouped by HDBSCAN* Cluster Labels") +
  theme_minimal() +
  facet_wrap(~hdbs_cluster, ncol=1, scale="free")
```



Question 21 compare the subfigures.

(a) Visually, do the genes in each cluster found by HDBSCAN in Item 20 correspond reasonably well to the associated functional category in the ground truth. The genes in each cluster labelled by HDBSCAN do in fact correspond reasonably well with their respective functional groups. The colour-coded cluster labels from the original Classe variable (ground truth) and the HDBSCAN classifications show results that correspond to the same trends observed in the contingency tables. Therefore, visually these genes do correspond well to their associated functional categories in the ground truth label.

(b) Look at the contingency table for the functional categories that have genes labelled as noise, then look at the corresponding pairs of sub-figures in Item 19 & 20, noticing that these genes are plotted in Item 19 but not in Item 20. Does this make each cluster visually clearer? The contingency table reveals that cluster 3 of the grount truth label was classified by the HDBSCAN algorithm to contain 55 outlier observations (label 0). This trend is observable following the blue lines of cluster 3 in figure 19 compared to the same coloured lines in label 2 of figure 20. The latter appears more organised and distinguishable, meaning the HDBSCAN algorithm successfully removed a significant number of outliers from this cluster. The same phenomenon is observed when comparing cluster 2 green lines from figure 19 with those of label 3 in figure 20. The latter appears significantly more distinct and bundled together. This is consistent with the observed outlier detection as highlighted by the contingency table. The identification of these outliers/noise do in fact make these clusters visually clearer as the lines are more bundled together.