Homework Assignment 3

Matthew Xu

25 November 2020

```
# knit from directory
setwd("/Users/MatthewXu/Desktop/PSTAT131")

library(readr)

knitr::opts_knit$set(root.dir = "/Users/MatthewXu/Desktop/PSTAT131")
```

```
library(tidyverse)
library(ROCR)
library(tree)
library(maptree)
library(class)
library(lattice)
library(ggridges)
library(superheat)
```

```
drug_use <- read_csv("drug.csv", col_names = c("ID", "Age", "Gender", "Education",
    "Country", "Ethnicity", "Nscore", "Escore", "Oscore", "Ascore", "Cscore", "Impulsive",
    "SS", "Alcohol", "Amphet", "Amyl", "Benzos", "Caff", "Cannabis", "Choc", "Coke",
    "Crack", "Ecstasy", "Heroin", "Ketamine", "Legalh", "LSD", "Meth", "Mushrooms",
    "Nicotine", "Semer", "VSA"))
```

```
drug_use <- drug_use %>% mutate_at(as.ordered, .vars = vars(Alcohol:VSA))
drug_use <- drug_use %>% mutate(Gender = factor(Gender, labels = c("Male", "Female"))) %>%
    mutate(Ethnicity = factor(Ethnicity, labels = c("Black", "Asian", "White", "Mixed:White/Bla
        "Other", "Mixed:White/Asian", "Mixed:Black/Asian"))) %>% mutate(Country = factor(Countr
    labels = c("Australia", "Canada", "New Zealand", "Other", "Ireland", "UK", "USA")))
```

1. "PROBLEM NUMBER ONE"

a. "PROBLEM NUMBER ONE, PART A"

```r
# mutate to factor yes or no if >= CL3
drug_use <- drug_use %>% mutate(recent_cannabis_use = factor(ifelse(drug_use$Cannabis >=
    "CL3", "Yes", "No"), labels = c("No", "Yes")))

# test to see if factor
class(drug_use$recent_cannabis_use)
```

```
## [1] "factor"
```

b. "PROBLEM NUMBER ONE, PART B"

```r
drug_use_subset <- drug_use %>% select(Age:SS, recent_cannabis_use)

# split into train.test sets of train set size = 1500
set.seed(1, sample.kind = "Rounding")
train.indices = sample(1:nrow(drug_use_subset), 1500)
drug_use_train = drug_use_subset[train.indices, ]
drug_use_test = drug_use_subset[-train.indices, ]

# dimensions of train should be 1500, test 385 (1835 - 1500)
dim(drug_use_train)
```

```
## [1] 1500   13
```

```r
dim(drug_use_test)
```

```
## [1] 385  13
```

c. "PROBLEM NUMBER ONE, PART C"

```r
# fitting logitisc regression to train data
glm.fit <- glm(drug_use_train$recent_cannabis_use ~ ., data = drug_use_train, family = binomial
summary(glm.fit)
```

```
##
## Call:
## glm(formula = drug_use_train$recent_cannabis_use ~ ., family = binomial,
##     data = drug_use_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0024  -0.5996   0.1512   0.5410   2.7525
##
## Coefficients:
```

```
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     1.33629    0.64895   2.059 0.039480 *
## Age                            -0.77441    0.09123  -8.489  < 2e-16 ***
## GenderFemale                   -0.65308    0.15756  -4.145 3.40e-05 ***
## Education                      -0.41192    0.08006  -5.145 2.67e-07 ***
## CountryCanada                  -0.67373    1.23497  -0.546 0.585377
## CountryNew Zealand             -1.24256    0.31946  -3.890 0.000100 ***
## CountryOther                    0.11062    0.49754   0.222 0.824056
## CountryIreland                 -0.50841    0.69084  -0.736 0.461773
## CountryUK                      -0.88941    0.39042  -2.278 0.022720 *
## CountryUSA                     -1.97561    0.20101  -9.828  < 2e-16 ***
## EthnicityAsian                 -1.19642    0.96794  -1.236 0.216443
## EthnicityWhite                  0.65189    0.63569   1.025 0.305130
## EthnicityMixed:White/Black      0.10814    1.07403   0.101 0.919799
## EthnicityOther                  0.66571    0.79791   0.834 0.404105
## EthnicityMixed:White/Asian      0.48986    0.96724   0.506 0.612535
## EthnicityMixed:Black/Asian     13.07740  466.45641   0.028 0.977634
## Nscore                         -0.08318    0.09163  -0.908 0.363956
## Escore                         -0.11130    0.09621  -1.157 0.247349
## Oscore                          0.64932    0.09259   7.013 2.33e-12 ***
## Ascore                          0.09697    0.08235   1.178 0.238990
## Cscore                         -0.30243    0.09179  -3.295 0.000984 ***
## Impulsive                      -0.14213    0.10381  -1.369 0.170958
## SS                              0.70960    0.11793   6.017 1.78e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2072.2  on 1499  degrees of freedom
## Residual deviance: 1185.4  on 1477  degrees of freedom
## AIC: 1231.4
##
## Number of Fisher Scoring iterations: 13
```

2. "PROBLEM NUMBER TWO"

```
# tree control
tree_parameters = tree.control(nobs = nrow(drug_use_train), minsize = 10, mindev = 0.001)
```

a. "PROBLEM NUMBER TWO, PART A"

```
nfold = 10
folds = seq.int(nrow(drug_use_train)) %>% ## sequential obs ids
cut(breaks = nfold, labels=FALSE) %>% ## sequential fold ids
```

```
sample ## random fold ids

drugtree <- tree(drug_use_train$recent_cannabis_use~., drug_use_train, control = tree_parameter

set.seed(1, sample.kind = "Rounding")
# K-Fold cross validation
#rand is number of cases, the fold partioning
cv = cv.tree(drugtree, rand = folds,  K=10, FUN=prune.misclass)

#find best tree size
#there are identical deviations for different sizes
#min of cv$size from the minimum positions of the minimum of cv$dev
cv$size
```

```
##  [1] 132  92  87  84  80  76  70  56  49  41  35  29  25  24  20  14  10   8   7
## [20]   5   4   2   1
```

```
cv$dev
```

```
##  [1] 315 315 315 315 315 315 315 315 315 315 315 315 315 315 315 315 315 316 323
## [20] 325 324 361 698
```

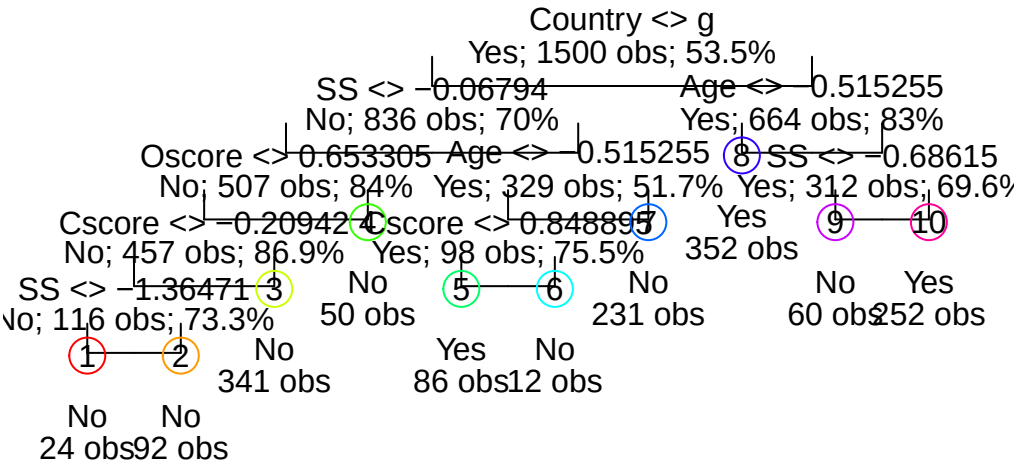
```
best_size = min(cv$size[which(cv$dev == min(cv$dev))])
best_size
```

```
## [1] 10
```

Best size of the pruned tree is of 8 internal nodes.

b. "PROBLEM NUMBER TWO, PART B"

```
# prune tree to best size
drugtree.pruned = prune.tree(drugtree, best = best_size)

# Plot the tree
draw.tree(drugtree.pruned, nodeinfo = TRUE)
title("Classification Tree Built on Training Set")
```



The first variable split in this decision tree is Country.

c. "PROBLEM NUMBER TWO, PART C"

```
# Predict on test set
tree.pred = predict(drugtree.pruned, drug_use_test, type = "class")

# confusion matrix for truths/falses
test_labels = drug_use_test$recent_cannabis_use
test.table = table(tree.pred, test_labels)
test.table
```

```
##          test_labels
## tree.pred  No Yes
##       No  163  52
##       Yes  25 145
```

```
# TPR = TP/(TP+FN) FPR = FP/(FP+TN)
TPR = test.table[2, 2]/(test.table[2, 2] + test.table[1, 2])
TPR
```

```
## [1] 0.7360406
```

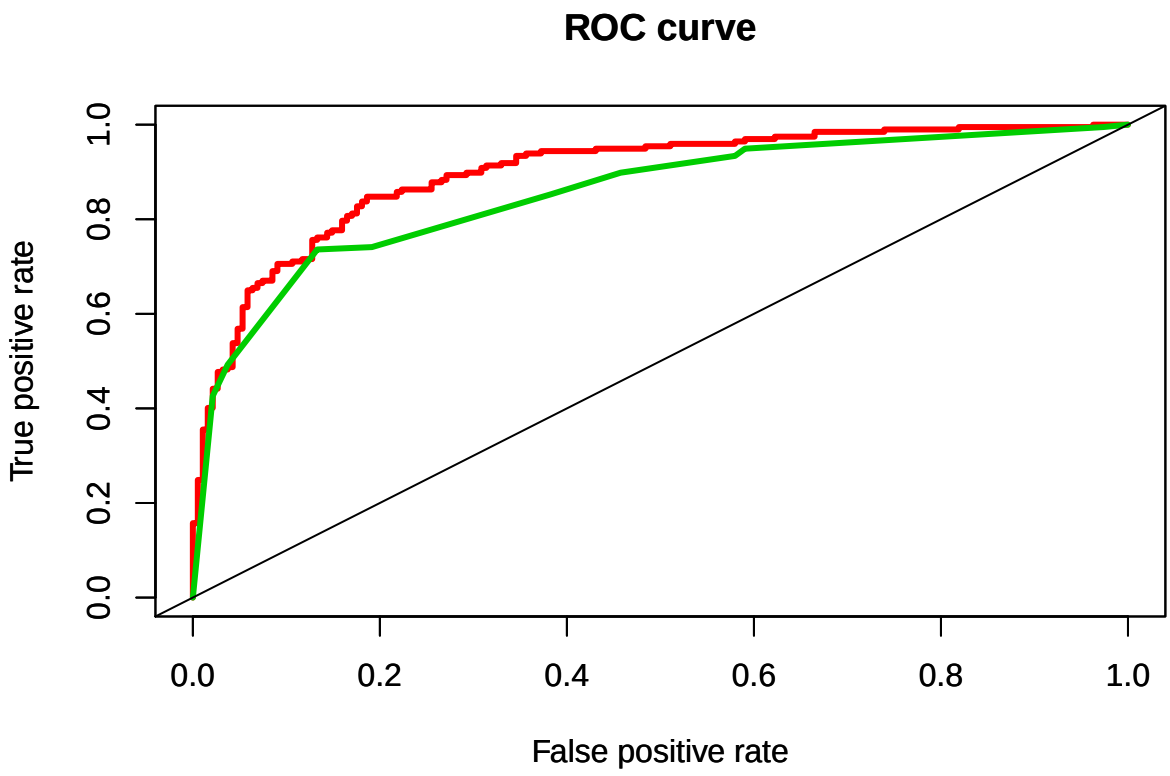
```
FPR = test.table[2, 1]/(test.table[2, 1] + test.table[1, 1])
FPR
```

```
## [1] 0.1329787
```

3. "PROBLEM NUMBER THREE"

a. "PROBLEM NUMBER THREE, PART A"

```
# prob training for logisitic regression fit model to test data
prob.training1 = predict(glm.fit, drug_use_test, type = "response")
# prob training for decision tree
prob.training2 = predict(drugtree.pruned, drug_use_test, type = "vector")

# First arument is the prob.training, second is true labels
LOGpred = prediction(prob.training1, drug_use_test$recent_cannabis_use)
DTpred = prediction(prob.training2[, 2], drug_use_test$recent_cannabis_use)

# We want TPR on the y axis and FPR on the x axis
perf1 = performance(LOGpred, measure = "tpr", x.measure = "fpr")
perf2 = performance(DTpred, measure = "tpr", x.measure = "fpr")

# plot both ROC curves on same plot
plot(perf1, col = 2, lwd = 3, main = "ROC curve")
par(new = TRUE)
plot(perf2, col = 3, lwd = 3, main = "ROC curve")
abline(0, 1)
```



b. "PROBLEM NUMBER THREE, PART B"

```
# Calculate AUC log
auc1 = performance(LOGpred, "auc")@y.values
auc1
```

```
## [[1]]
## [1] 0.8973971
```

```
# Calculate AUC Descision Tree
auc2 = performance(DTpred, "auc")@y.values
auc2
```

```
## [[1]]
## [1] 0.8526299
```

The logistic Regression model has the better AUC value.

4. "PROBLEM NUMBER FOUR"

```
leukemia_data <- read_csv("leukemia_data.csv")
```

a. "PROBLEM NUMBER FOUR, PART A"

```
# change variable Type to factor
leukemia_data <- leukemia_data %>% mutate_at(vars("Type"), as.factor)

# check to see if factor
class(leukemia_data$Type)
```

```
## [1] "factor"
```

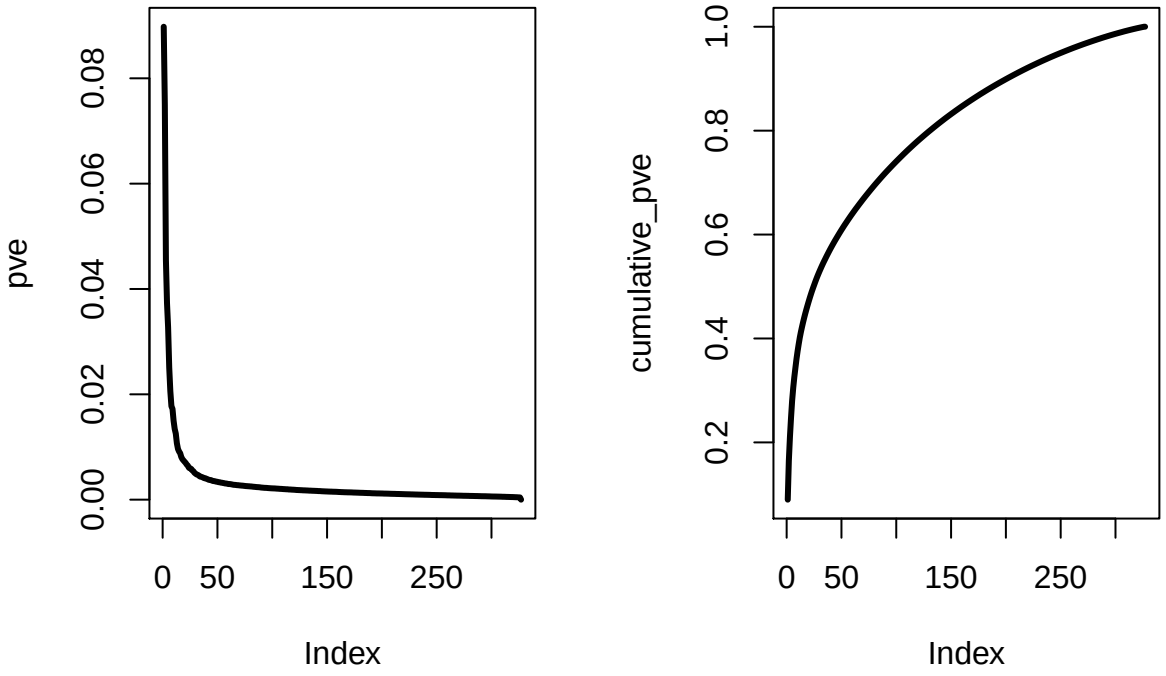
```
# frequency table for each leukemia subtype
table(leukemia_data$Type)
```

```
##
##    BCR-ABL   E2A-PBX1 Hyperdip50        MLL     OTHERS      T-ALL   TEL-AML1
##         15         27         64         20         79         43         79
```

The BCR-ABL leukemia subtype occurs the least.

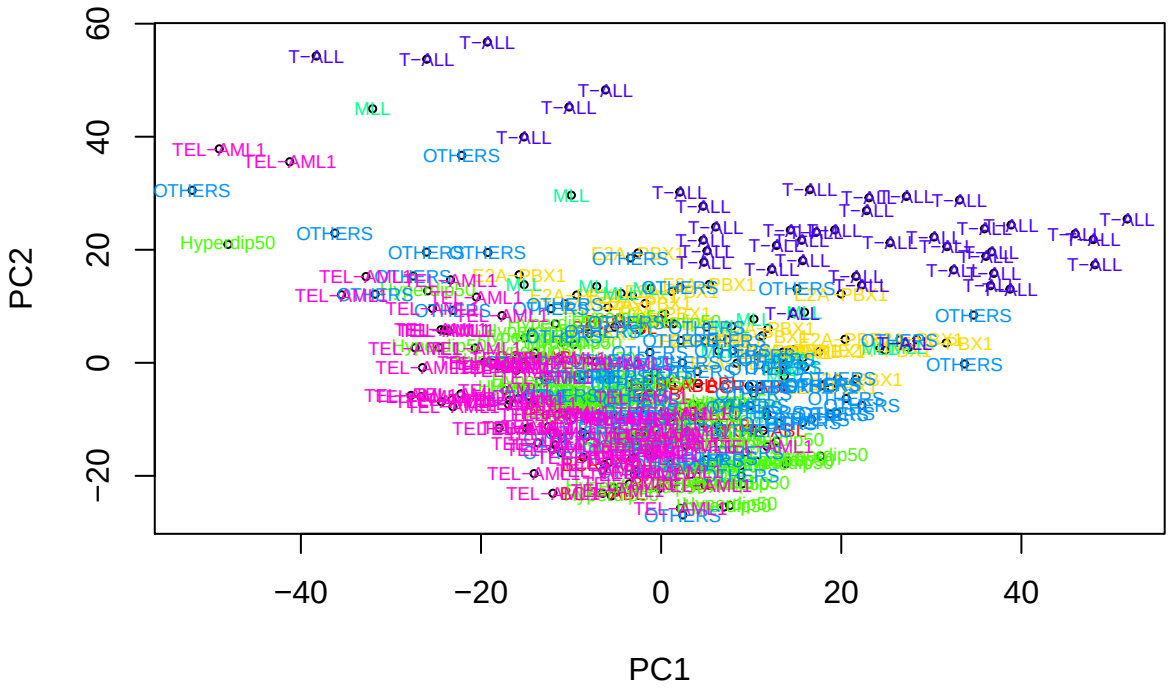
b. "PROBLEM NUMBER FOUR, PART B"

```r
# using prcomp to calculate pve and cumulative pve
leukemia_pca <- prcomp(leukemia_data[, -c(1)], scale = TRUE, center = TRUE)
sdev <- leukemia_pca$sdev
pve <- sdev^2/sum(sdev^2)
cumulative_pve <- cumsum(pve)

## This will put the next two plots side by side
par(mfrow = c(1, 2))

## Plot proportion of variance explained \
plot(pve, type = "l", lwd = 3)
plot(cumulative_pve, type = "l", lwd = 3)
```



c. "PROBLEM NUMBER FOUR, PART C"

```r
# load colors
require(graphics)
rainbow_colors <- rainbow(7)
plot_colors <- rainbow_colors[leukemia_data$Type]

new_coords <- leukemia_pca$x[, 1:2]
plot(new_coords, cex = 0.5)
# from piazza do not need -new_coords
text(new_coords, labels = leukemia_data$Type, col = plot_colors, cex = 0.6)
```




```
sorted_PC1 <- sort(abs(leukemia_pca$x[, 1]), decreasing = TRUE)
head(sorted_PC1, n = 6)
```

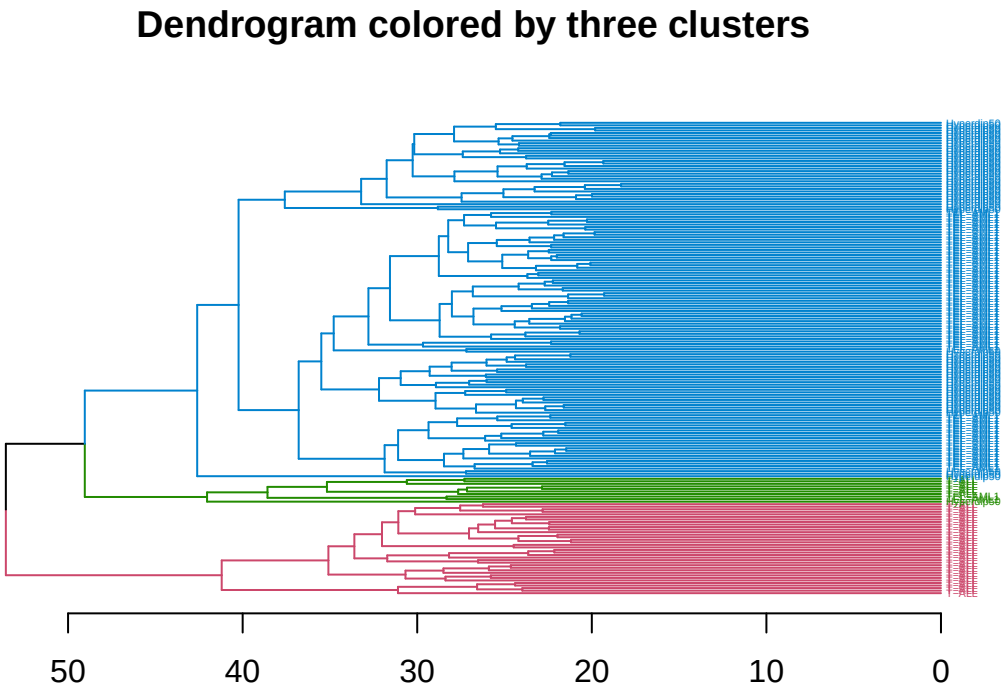
```
## [1] 52.06836 51.75384 49.06527 48.16035 48.09339 47.96769
```

Along the PC1 axis, the group that appears to be the most seperated from the others is the T-All gene. The six genes from PC1 that have the largest weighted have weights of 52.06836, 51.75384, 49.06527, 48.16035, 48.09339, 47.96769.

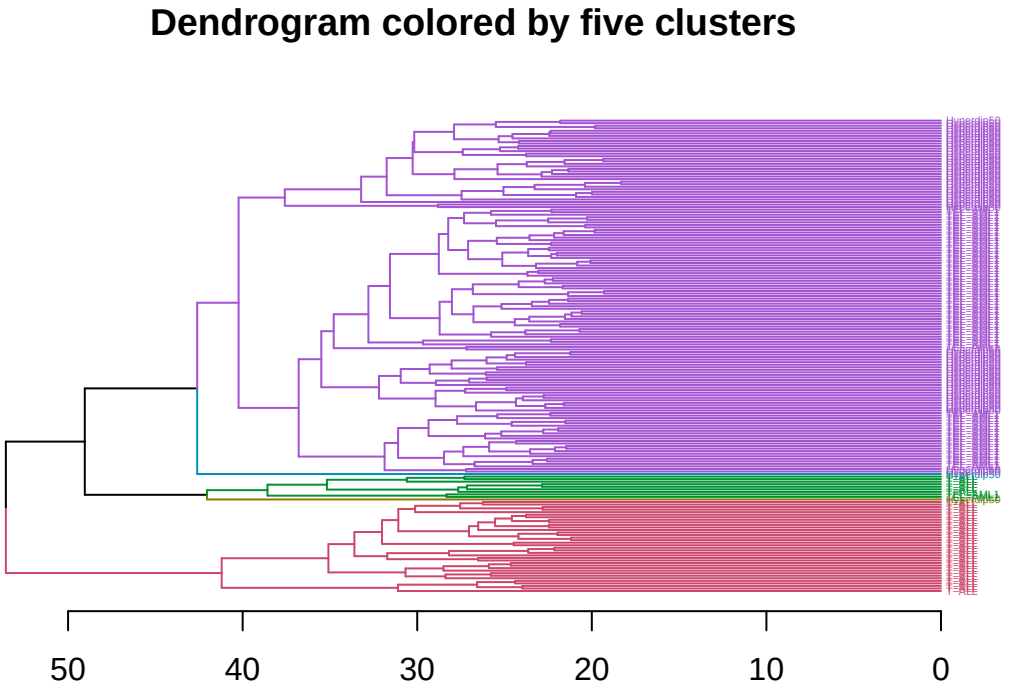
f. “PROBLEM NUMBER FOUR, PART F”

```
library(dendextend)

# subset data to these 3 leukemia subtypes
leukemia_subset = filter(leukemia_data, Type == "T-ALL" | Type == "TEL-AML1" | Type ==
    "Hyperdip50")

# eucledian distance with compltete linkage
dis = dist(leukemia_subset, method = "euclidean")
set.seed(213)
leukemia.hc = hclust(dis, method = "complete")

## dendrogram: branches colored by 3 groups
dend1 = as.dendrogram(leukemia.hc)
# color branches and labels by 3 clusters
dend1 = color_branches(dend1, k = 3)
```

```
dend1 = color_labels(dend1, k = 3)
# change label size
dend1 = set(dend1, "labels_cex", 0.3)

# add true labels to observations
dend1 = set_labels(dend1, labels = leukemia_subset$Type[order.dendrogram(dend1)])  # plot the
plot(dend1, horiz = T, main = "Dendrogram colored by three clusters")
```




```
## dendrogram: branches colored by 5 groups
dend2 = as.dendrogram(leukemia.hc)
# color branches and labels by 5 clusters
dend2 = color_branches(dend2, k = 5)
dend2 = color_labels(dend2, k = 5)
# change label size
dend2 = set(dend2, "labels_cex", 0.3)

# add true labels to observations
dend2 = set_labels(dend2, labels = leukemia_subset$Type[order.dendrogram(dend2)])  # plot the
plot(dend2, horiz = T, main = "Dendrogram colored by five clusters")
```


Dendrogram colored by five clusters
50
40
30
20
10
0