HW3

Justin Lee

11/16/2020

```
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'))
```

(1)

```
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/Black", "Other",
"Mixed:White/Asian",
"Mixed:Black/Asian"))) %>%
mutate(Country = factor(Country, labels=c("Australia", "Canada", "New Zealand",
"Other", "Ireland", "UK", "USA")))
```

(a)

```
drug_use <- drug_use %>%
  mutate(recent_cannabis_use = factor(ifelse(Cannabis >= "CL3", "Yes", "No"),
                                      levels = c("No","Yes")))
#Check to see if the new column exists
names(drug_use)
```

```
##  [1] "ID"                 "Age"                "Gender"
##  [4] "Education"          "Country"            "Ethnicity"
##  [7] "Nscore"             "Escore"             "Oscore"
## [10] "Ascore"             "Cscore"             "Impulsive"
## [13] "SS"                 "Alcohol"            "Amphet"
## [16] "Amyl"               "Benzos"             "Caff"
## [19] "Cannabis"           "Choc"               "Coke"
## [22] "Crack"              "Ecstasy"            "Heroin"
## [25] "Ketamine"           "Legalh"             "LSD"
## [28] "Meth"               "Mushrooms"          "Nicotine"
## [31] "Semer"              "VSA"                "recent_cannabis_use"
```

(b)

```
set.seed(123)
drug_use_subset <- drug_use %>% select(Age:SS, recent_cannabis_use)
drug_use_subset
```

```
## # A tibble: 1,885 x 13
##        Age Gender Education Country Ethnicity Nscore   Escore  Oscore Ascore
##      <dbl> <fct>      <dbl> <fct>   <fct>      <dbl>    <dbl>   <dbl>  <dbl>
##  1  0.498  Female   -0.0592 USA     Mixed:Wh~  0.313 -0.575   -0.583  -0.917
##  2 -0.0785 Male      1.98   USA     White     -0.678  1.94     1.44    0.761
##  3  0.498  Male     -0.0592 USA     White     -0.467  0.805   -0.847  -1.62
##  4 -0.952  Female    1.16   USA     White     -0.149 -0.806   -0.0193  0.590
##  5  0.498  Female    1.98   USA     White      0.735 -1.63    -0.452  -0.302
##  6  2.59   Female   -1.23   UK      White     -0.678 -0.300   -1.56    2.04
##  7  1.09   Male      1.16   Austra~ White     -0.467 -1.09    -0.452  -0.302
##  8  0.498  Male     -1.74   USA     White     -1.33   1.94    -0.847  -0.302
##  9  0.498  Female   -0.0592 UK      White      0.630  2.57    -0.976   0.761
## 10  1.82   Male      1.16   USA     White     -0.246  0.00332 -1.42    0.590
## # ... with 1,875 more rows, and 4 more variables: Cscore <dbl>,
## #   Impulsive <dbl>, SS <dbl>, recent_cannabis_use <fct>
```

```
#Train and Test
train_index = sample(nrow(drug_use_subset), 1500)

drug_use_train = drug_use_subset[train_index, ]
drug_use_test = drug_use_subset[-train_index, ]
dim(drug_use_train)
```

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

```
dim(drug_use_test)
```

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

The dimensions of the training set is 1500 along with 385 in the test set which comes out to 1885 which verifies the data set is the right size.

(c)

```
drug_train_fit= glm(recent_cannabis_use~ ., data = drug_use_train, family = binomial)

drug_train_predict = predict(drug_train_fit, type = "response")
summary(drug_train_predict)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.005798 0.182827 0.538137 0.536667 0.906204 1.000000
```

(2)

```
tree_parameters = tree.control(nobs=nrow(drug_use_train), minsize=10, mindev=1e-3)
```

(a)

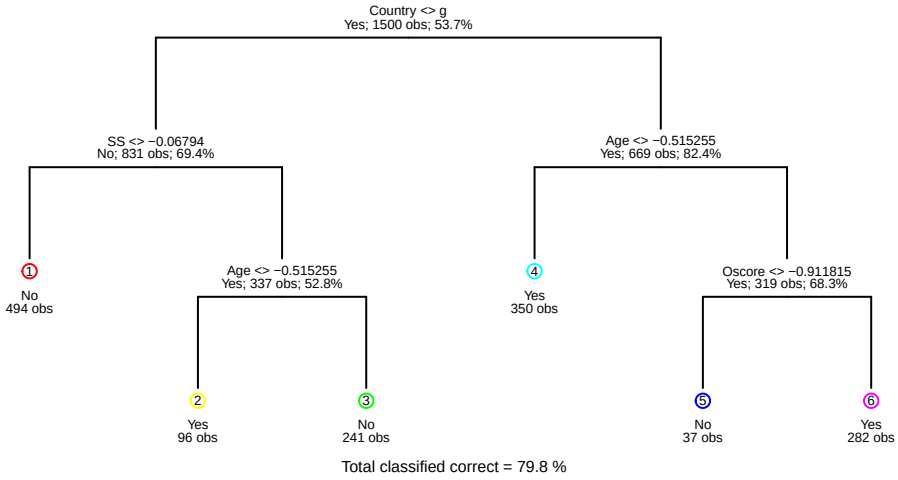
```
set.seed(123)
drug_use_tree = tree(recent_cannabis_use~., data = drug_use_train, control = tree_parameters)
drugtree = cv.tree(drug_use_tree, FUN = prune.misclass, K = 10)
devsize = as.data.frame(cbind(drugtree$size, drugtree$dev))
devsize = devsize[order(devsize$V1),]
best_size = devsize$V1[which.min(devsize$V2)]
best_size
```

```
## [1] 6
```

We can see from our model that the size of the tree that minimizes the cross validation error is 6.

(b)

```
drug_pruned = prune.tree(drug_use_tree, best = best_size, method = "misclass")
draw.tree(drug_pruned, cex = 0.4, nodeinfo = TRUE)
```



The first split in the decision tree is the variable "Country".

(c)

```
drug_pred = predict(drug_pruned, drug_use_test, type = "class")
confusion_test = table(predicted = drug_pred, true = drug_use_test$recent_cannabis_use)
confusion_test
```

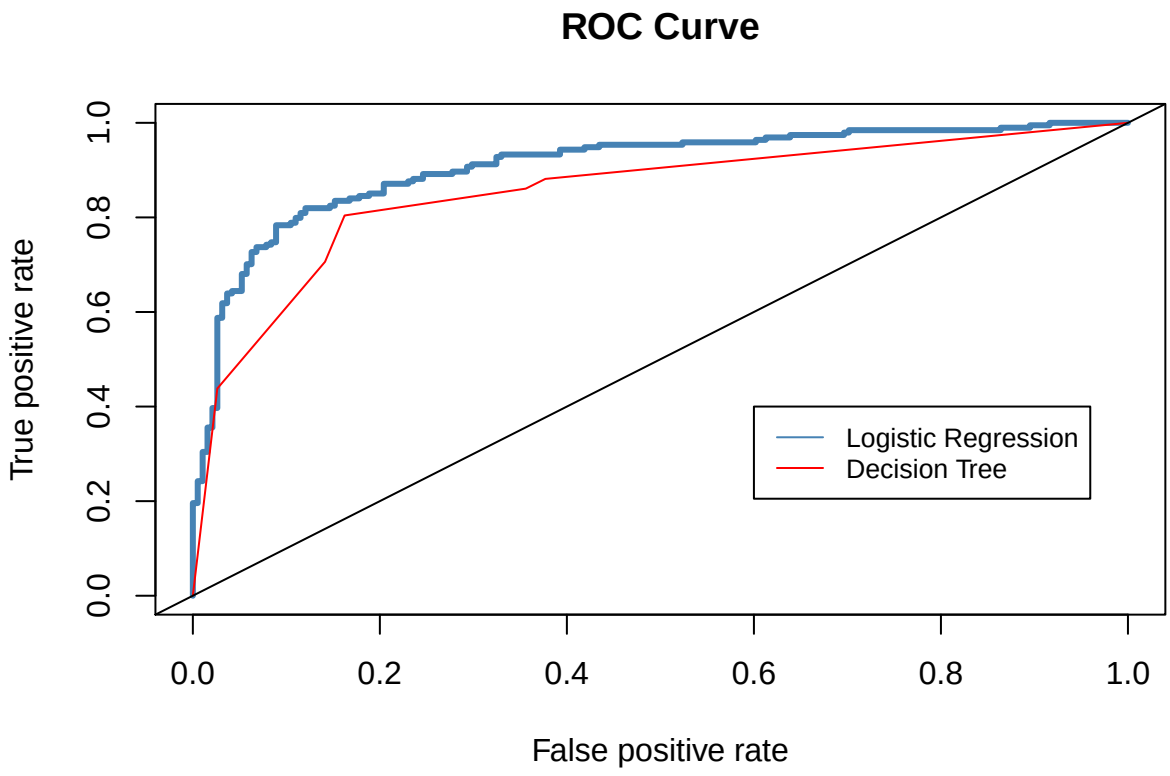
```
##          true
## predicted  No Yes
##       No  160  38
##       Yes  31 156
```

The equation of TPR is given as $\frac{TP}{TP+FN}$ and FPR as $\frac{FP}{FP+TN}$.
TPR $= \frac{160}{160+31} = 0.8376963$
FPR $= \frac{38}{38+156} = 0.1958763$

(3)

(a)

```
#Logistic
drug_test_log_predict = predict(drug_train_fit, drug_use_test, type = "response")
predLog = prediction(drug_test_log_predict, drug_use_test$recent_cannabis_use)
perfLog = performance(predLog, measure = "tpr", x.measure = "fpr")
plot(perfLog, col = "steelblue", lwd = 3, main = "ROC Curve")


#Decision
drug_test_predict = predict(drug_pruned, drug_use_test, type = "vector")
predDec = prediction(drug_test_predict[,2], drug_use_test$recent_cannabis_use)
perfDec = performance(predDec, measure = "tpr", x.measure = "fpr")
plot(perfDec, add = TRUE, col = "red")
abline(0,1)
legend(0.6,0.4, legend = c("Logistic Regression", "Decision Tree"), col = c("steelblue", "red"), lty =
```



(b)

```
log_auc = performance(predLog, measure = "auc")
log_auc = log_auc@y.values[[1]]
log_auc
```

```
## [1] 0.908323
```

```
dec_auc = performance(predDec, measure = "auc")
dec_auc = dec_auc@y.values[[1]]
dec_auc
```

```
## [1] 0.8530658
```

From the calculations shown above, the logistic regression model gives us an AUC of 0.908323 and the decision tree model gives us an AUC of 0.8530658. We can clearly see that the logistic regression model has a larger AUC.

(4)

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

(a)

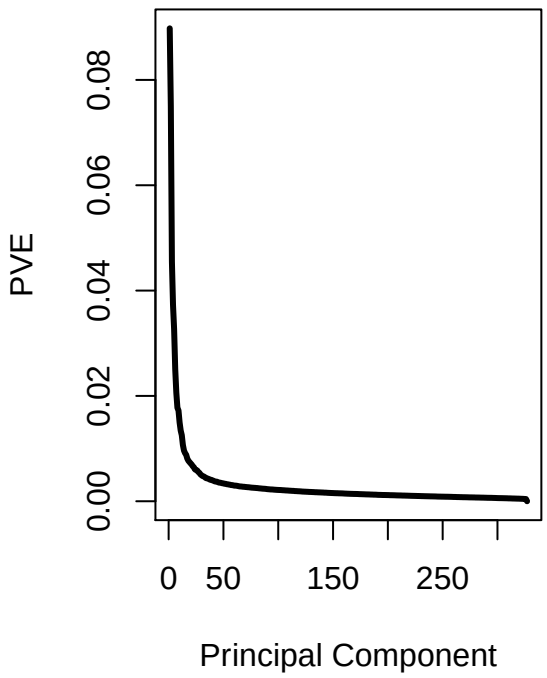
```
leukemia_data = leukemia_data %>%
  mutate(Type = factor(Type))
table(leukemia_data$Type)
```

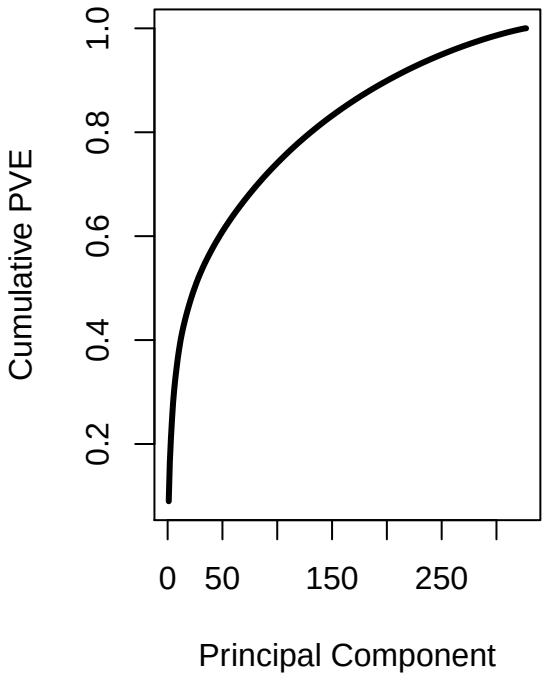
```
##
##    BCR-ABL   E2A-PBX1 Hyperdip50        MLL     OTHERS      T-ALL   TEL-AML1
##         15         27         64         20         79         43         79
```

We can see here that the BCR-ABL is the subtype that occurs the least in this data

(b)

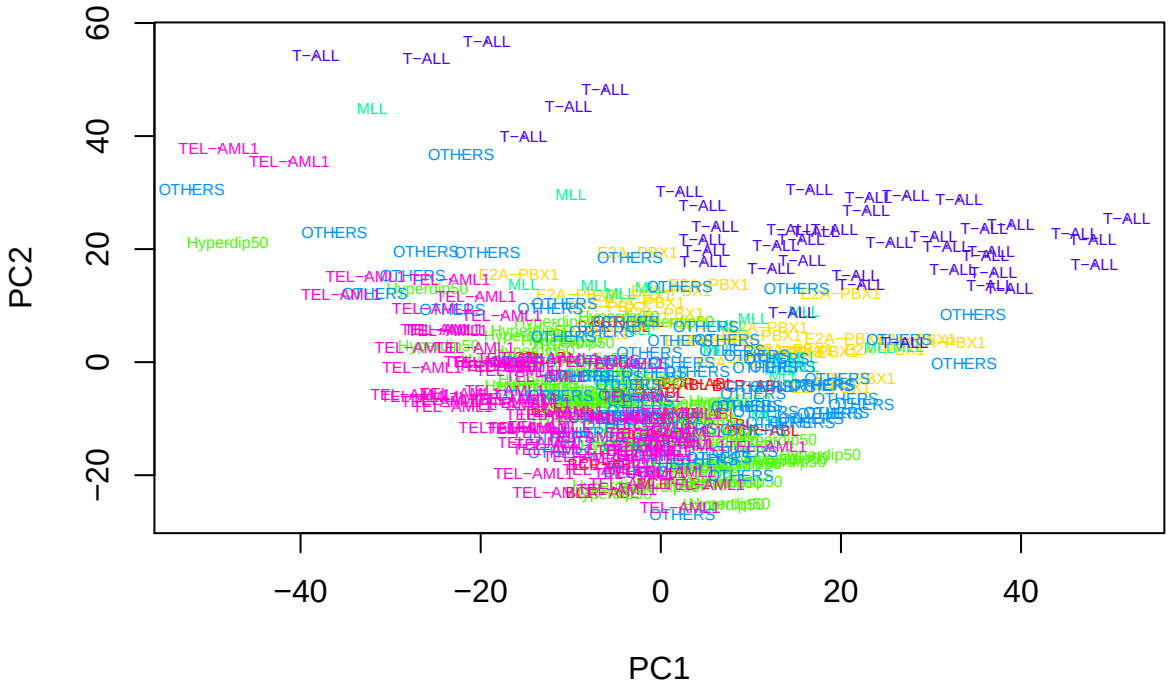
```
#Setup for pve
pr_out = prcomp(subset(leukemia_data, select = -c(Type)), scale = TRUE)
pr_var = pr_out$sdev^2


pve <- pr_var / sum(pr_var)
cumulative_pve <- cumsum(pve)


par(mfrow=c(1, 2))
plot(pve, type="l", lwd=3, xlab = "Principal Component", ylab = "PVE")
plot(cumulative_pve, type="l", lwd=3, xlab = "Principal Component", ylab = "Cumulative PVE")
```





(c)

```
#ScatterPlot
rainbow_colors <- rainbow(7)
plot_colors <- rainbow_colors[leukemia_data$Type]
plot(pr_out$x, col = plot_colors, cex = 0.001)
text(pr_out$x, labels = leukemia_data$Type, col = plot_colors, cex = 0.5)
```




```
#Second part
head(sort(abs(pr_out$rotation[,1])), 6)
```

```
##        SRSF8        BUB1B       SEC11A     35985_at        EVI2B       ZFAND5
## 7.950999e-07 3.499181e-06 2.400636e-05 3.193166e-05 3.282533e-05 3.513191e-05
```

The group that is most clearly separated from the PC1 axis is *TEL_AML1*. The genes with the highest absolute loadings for PC1 is *SRSF8, BUB1B, SEC11A, 35985_at, EVI2B, ZFAND5*.

(f)

```
library(dendextend)
```

```
##
## ---------------------
## Welcome to dendextend version 1.14.0
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## Or contact: <tal.galili@gmail.com>
```

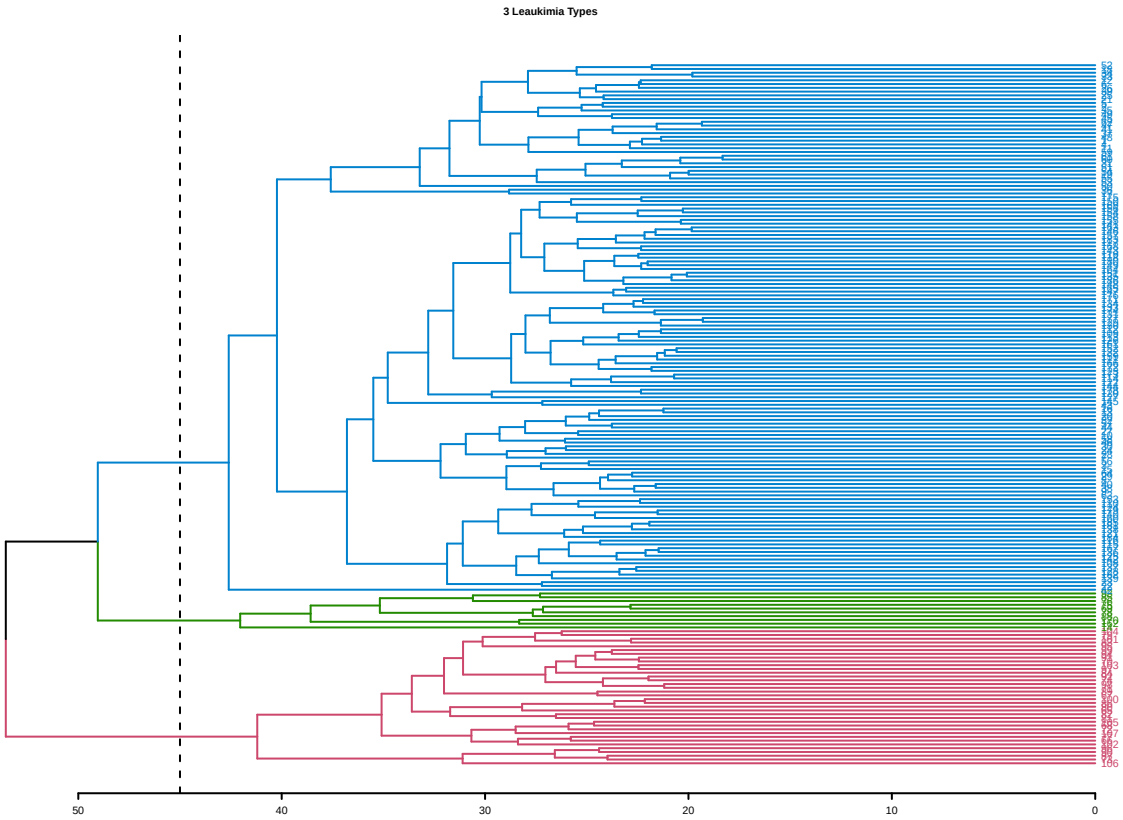
```
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

```
##
## Attaching package: 'dendextend'
```

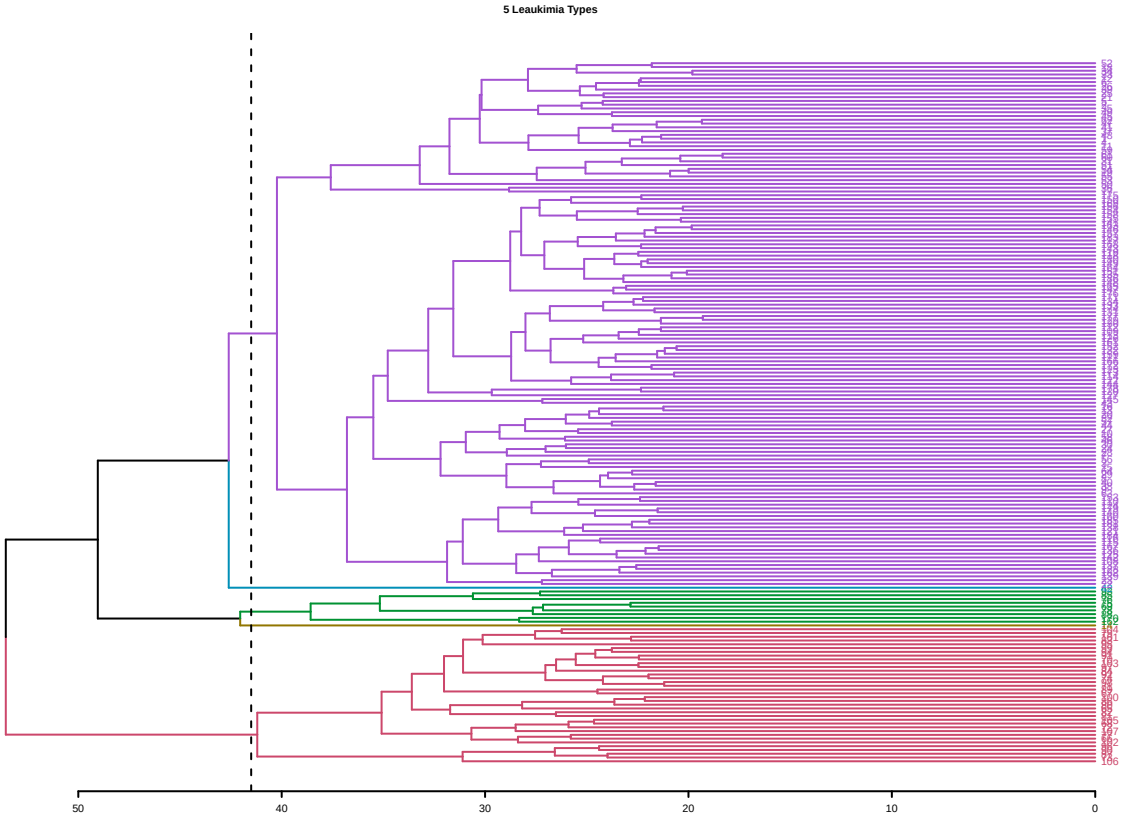
```
## The following object is masked from 'package:rpart':
##
##     prune
```

```
## The following object is masked from 'package:stats':
##
##     cutree
```

```
leukemia_subset = filter(leukemia_data, leukemia_data$Type == 'T-ALL' | leukemia_data$Type == 'TEL-AML1
leuk_dist = dist(leukemia_subset)
set.seed(123)
leuk_Hclust = hclust(leuk_dist)

#First plot
dend = as.dendrogram(leuk_Hclust)
d3 = color_branches(dend, k = 3)
dat = color_labels(d3, k = 3)
par(cex = 0.3)
plot(dat, horiz = TRUE, main = "3 Leaukimia Types")
abline(v = 45, lty = 2)
```




```
#Second plot
d5 = color_branches(dend, k = 5)
dat2 = color_labels(d5, k = 5)
par(cex = 0.3)
plot(dat2, horiz = TRUE, main = "5 Leaukimia Types")
abline(v = 41.5, lty = 2)
```




```
cutree(dend, k = 5)
```

```
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   1   1   1   1   1   1   1   1   1   1   1   1   1   2   1   1   1   1   1   1
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   1   3   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   1   1   1   1   4   4   4   4   5   4   4   4   4   4   5   5   4   5   4   4
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   4   4   4   4   5   4   4   5   4   4   4   4   5   4   4   4   4   4   4   4
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   4   4   4   4   4   4   4   1   1   1   1   1   1   1   1   1   1   1   1   1
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   1   5   1   1   1   1   1   1   1   5   1   1   1   1   1   1   1   1   1   1
## 181 182 183 184 185 186
##   1   1   1   1   1   1
```