Homework Assignment 3

Marissa Santiago

May 21, 2021

```
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','
```

Problem 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", "
```

1a

```
drug_use <- drug_use %>% mutate(recent_cannabis_use=factor(ifelse(Cannabis>= 'CL3','Yes','No'),levels=c
class(drug_use$recent_cannabis_use)
```

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

```
#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"
```

1b

```
drug_use_subset <- drug_use %>% select(Age:SS, recent_cannabis_use)
#randomly sample to split data into training set and test set
set.seed(1)
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,]

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.

1c

```
glm.fit <- glm(recent_cannabis_use ~ ., data = drug_use_train, family = binomial(link = "logit"))
summary(glm.fit)
```

```
##
## Call:
## glm(formula = recent_cannabis_use ~ ., family = binomial(link = "logit"),
##     data = drug_use_train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.907  -0.597   0.142   0.543   2.660
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.9495     0.6457    1.47  0.14146
## Age                        -0.8441     0.0933   -9.05  < 2e-16 ***
```

```
## GenderFemale                -0.5593     0.1571   -3.56  0.00037 ***
## Education                   -0.3339     0.0796   -4.19  2.7e-05 ***
## CountryCanada               13.1090   627.2275    0.02  0.98333
## CountryNew Zealand          -1.1684     0.3185   -3.67  0.00024 ***
## CountryOther                -0.0568     0.4677   -0.12  0.90341
## CountryIreland              -0.2876     0.6757   -0.43  0.67035
## CountryUK                   -0.4337     0.3704   -1.17  0.24167
## CountryUSA                  -1.7564     0.1926   -9.12  < 2e-16 ***
## EthnicityAsian              -0.6703     0.9604   -0.70  0.48523
## EthnicityWhite               0.7405     0.6384    1.16  0.24608
## EthnicityMixed:White/Black  -0.0471     1.0901   -0.04  0.96551
## EthnicityOther               1.0789     0.7682    1.40  0.16021
## EthnicityMixed:White/Asian   0.7253     1.0156    0.71  0.47518
## EthnicityMixed:Black/Asian  14.2715   766.2817    0.02  0.98514
## Nscore                      -0.1014     0.0903   -1.12  0.26155
## Escore                      -0.1338     0.0956   -1.40  0.16174
## Oscore                       0.7100     0.0914    7.77  7.8e-15 ***
## Ascore                       0.0306     0.0823    0.37  0.71025
## Cscore                      -0.3585     0.0913   -3.93  8.6e-05 ***
## Impulsive                   -0.0904     0.1009   -0.90  0.37029
## SS                           0.5807     0.1084    5.36  8.4e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2077.2  on 1499  degrees of freedom
## Residual deviance: 1202.1  on 1477  degrees of freedom
## AIC: 1248
##
## Number of Fisher Scoring iterations: 14
```

Problem 2

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

2a

```
#Use 10-fold CV to select the a tree which minimizes the cross-validation misclassification rate.
set.seed(1)
tree.drug_use = tree(recent_cannabis_use~., data = drug_use_train, control = tree_parameters)
cv = cv.tree(tree.drug_use, FUN = prune.misclass, K = 10)

#get the indices for tree of smallest misclassification rate
best_size.indices = cv$dev %>% which(x = (. == min(.)))

#calculate missclass rate
best_size.misclass <- cv$dev[best_size.indices] %>% min
best_size.misclass
```

```
## [1] 314
```

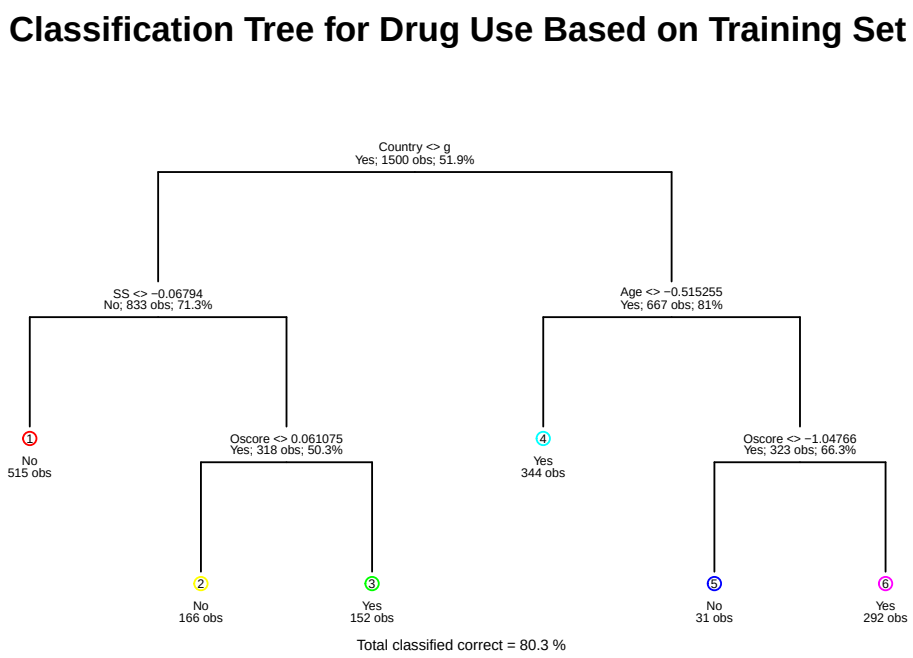
```
#actual
best_size <- cv$size[best_size.indices] %>% min
best_size
```

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

We can see from our model the size of the tree that minimizes the cross validation error is 6 with a minimum rate of 314.

2b

```
#prune tree to best size
pruned.drug_use = prune.misclass(tree.drug_use, best = best_size)
draw.tree(pruned.drug_use, cex = 0.4, nodeinfo = TRUE)
title("Classification Tree for Drug Use Based on Training Set")
```



The first variable that is split is 'Country'.

2c

```r
set.seed(1)
#truth is the response values
truth <- drug_use_test$recent_cannabis_use
#test.drug_use <- tree(recent_cannabis_use~., drug_use_test, control=tree_parameters)
prediction <- predict(object = pruned.drug_use, newdata = drug_use_test %>% select(-recent_cannabis_use
confusion_test = table(prediction, truth)
confusion_test
```

```
##           truth
## prediction  No Yes
##        No  125  45
##        Yes  40 175
```

```r
#calc TPR and FPR
tpr <- confusion_test[2,2]/((confusion_test[2,2] + confusion_test[1,2]))
tpr
```

```
## [1] 0.7955
```

```r
fpr <- confusion_test[2,1]/((confusion_test[2,1] + confusion_test[1,1]))
fpr
```

```
## [1] 0.2424
```

The equation of TPR is given as $\frac{TP}{TP+FN}$ and FPR as $\frac{FP}{FP+TN}$.

```r
cat('TPR is', tpr, '\n')
```

```
## TPR is 0.7955
```

```r
cat('FPR is', fpr)
```

```
## FPR is 0.2424
```

Problem 3

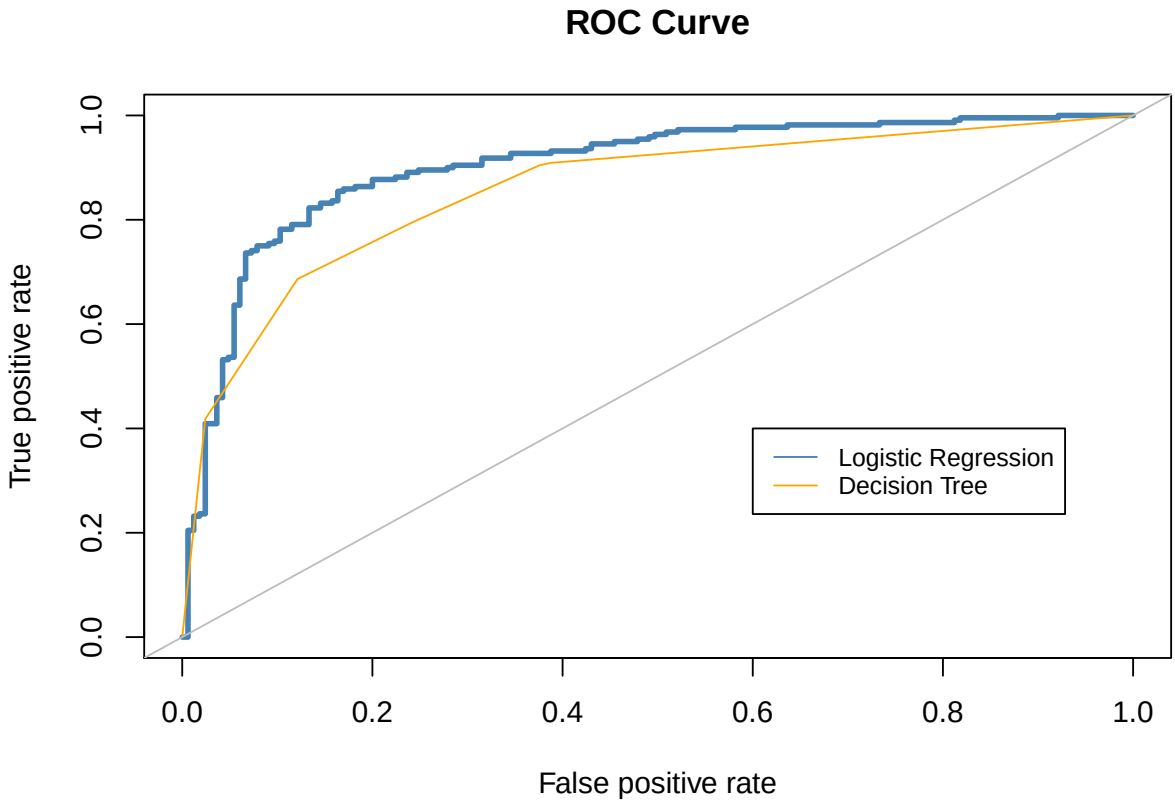
3a

```r
#Logistic
drug_test_log_predict = predict(glm.fit, drug_use_test, type = "response")
predLogistic =  prediction(drug_test_log_predict, drug_use_test$recent_cannabis_use)
perfLogistic = performance(predLogistic, measure = "tpr", x.measure = "fpr")
plot(perfLogistic, col = "steelblue", lwd = 3, main = "ROC Curve")

#Decision Tree
drug_test_predict = predict(pruned.drug_use, drug_use_test, type = "vector")
```

```
predDecision = prediction(drug_test_predict[,2], drug_use_test$recent_cannabis_use)
perfDecision<- performance(predDecision, measure="tpr", x.measure="fpr")
plot(perfDecision, add=TRUE, col = "orange")
abline(0,1, col = "grey")

legend(0.6,0.4, legend = c("Logistic Regression", "Decision Tree"), col = c("steelblue", "orange"), lty
```



3b

```
#Compute the AUC
auc_log = performance(predLogistic, "auc")@y.values[[1]]
auc_dec = performance(predDecision, "auc")@y.values[[1]]
cat("The AUC of logistic is", as.numeric(auc_log), '\n')
```

```
## The AUC of logistic is 0.9026
```

```
cat("The AUC of decision tree is", as.numeric(auc_dec))
```

```
## The AUC of decision tree is 0.8571
```

We can see that the logistic regression model has a larger AUC.

Problem 4

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

```
## Warning: Duplicated column names deduplicated: 'FCGRT' => 'FCGRT_1' [3],
## 'TUBB4B' => 'TUBB4B_1' [49], 'SSR1' => 'SSR1_1' [67], 'HSP90AB1' =>
## 'HSP90AB1_1' [115], 'TMBIM6' => 'TMBIM6_1' [118], 'GAB1' => 'GAB1_1' [119],
## 'MPHOSPH9' => 'MPHOSPH9_1' [153], 'STK38' => 'STK38_1' [157], 'SFPQ' =>
## 'SFPQ_1' [159], 'RIPOR2' => 'RIPOR2_1' [181], 'HLA-F' => 'HLA-F_1' [188],
## 'PRPF40A' => 'PRPF40A_1' [198], 'SEPT6' => 'SEPT6_1' [205], 'CD22' =>
## 'CD22_1' [235], 'NCF4' => 'NCF4_1' [250]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   Type = col_character()
## )
## i Use `spec()` for the full column specifications.
```

4a

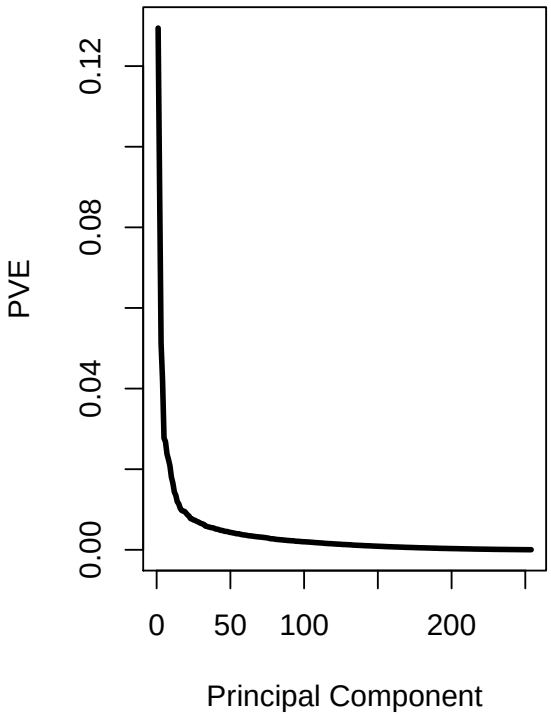
```
leukemia_data = leukemia_data %>%
  mutate(Type = factor(Type))
table(leukemia_data$Type)
```

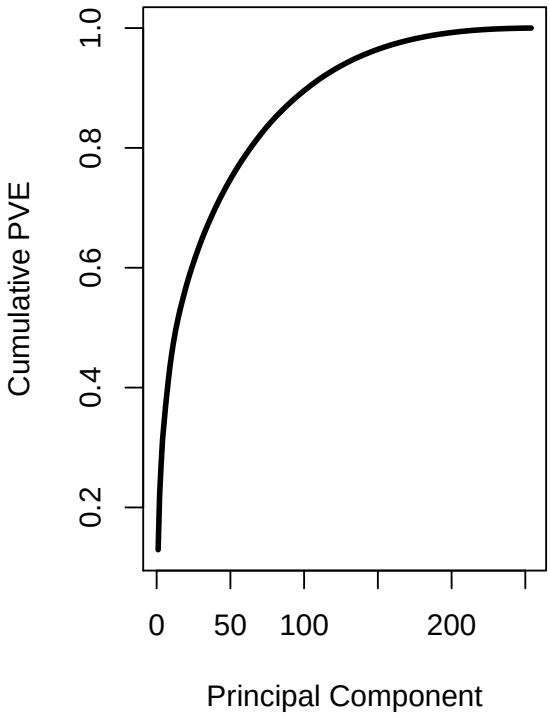
```
##
##    BCR-ABL   E2A-PBX1 Hyperdip50        MLL     OTHERS      T-ALL   TEL-AML1
##         15         27         64         20         79         43         79
```

BCR-ABL is the subtype that occurs the least in this data.

4b

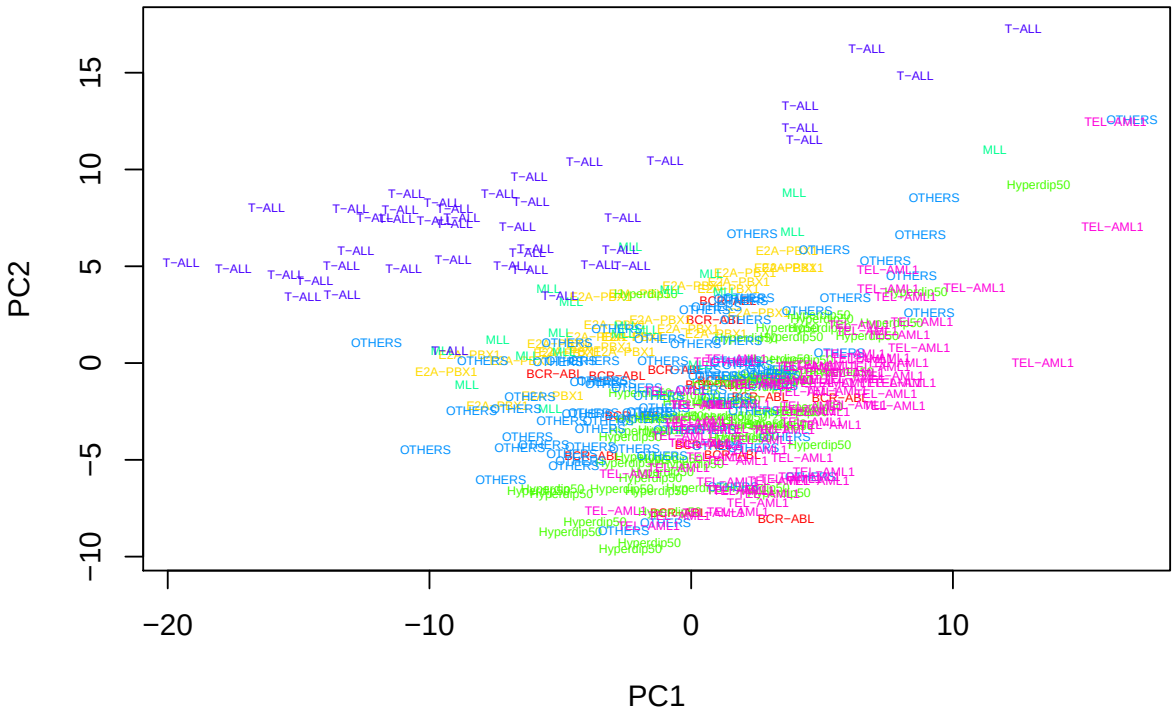
```
leukemia <- leukemia_data %>% select(-Type)
pca = prcomp(leukemia, scale = TRUE, center = TRUE)
pca_var = pca$sdev^2
pve <- pca_var / sum(pca_var)
cumulative_pve <- cumsum(pve)
par(mfrow=c(1, 2))
plot(pve, type="l", lwd=3, xlab = "Principal Component", ylab = "PVE")
plot(cumulative_pve, type="l", lwd=3, xlab = "Principal Component", ylab = "Cumulative PVE")
```





4c

```
#ask friend
rainbow_colors <- rainbow(7)
plot_colors <- rainbow_colors[leukemia_data$Type]
plot(pca$x, col=plot_colors, cex=0)
text(pca$x, labels=leukemia_data$Type, col=plot_colors, cex=0.4)
```




```
head(sort(abs(pca$rotation[,1])), 6)
```

```
##     ACAP1  RNASEH2B     SUMO4      MDM1     SRP72  31503_at
## 9.009e-05 2.252e-04 8.463e-04 1.241e-03 1.409e-03 2.205e-03
```

The group that is most clearly separated from others along the PC1 axis is *T-ALL*. The genes with the highest absolute loadings for PC1 is *SRSF8, BUB1B, SEC11A, 35985_at, EVI2B, and ZFAND5*.

4f

```
library(dendextend)
```

```
##
## ---------------------
## Welcome to dendextend version 1.15.1
## 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
```

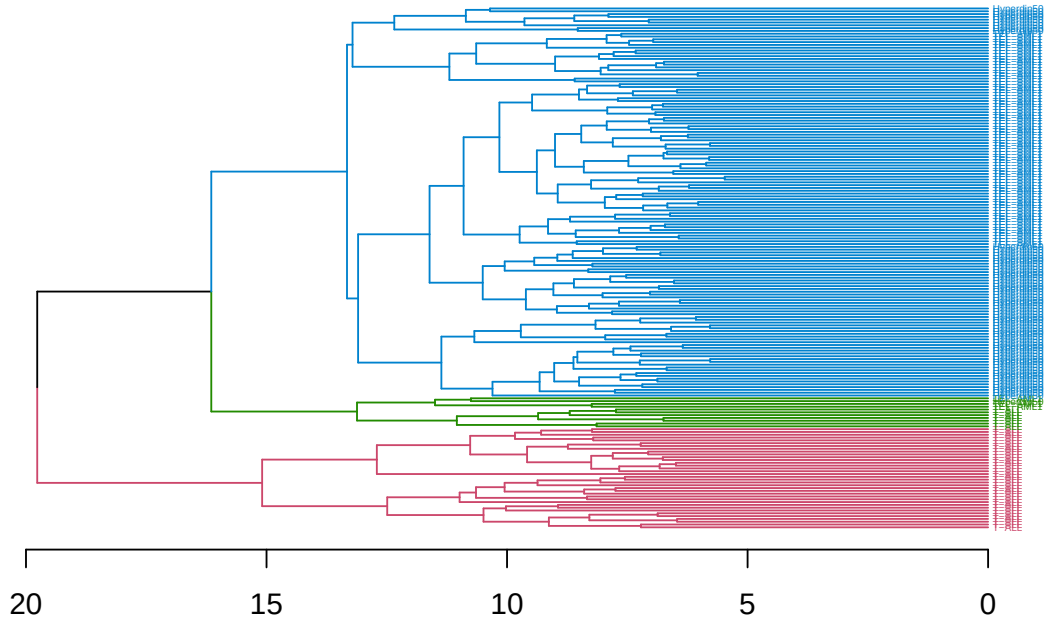
```
## The following object is masked from 'package:stats':
##
##     cutree
```

```
set.seed(1)
leukemia_subset <- filter(leukemia_data, leukemia_data$Type == 'T-ALL' | leukemia_data$Type == 'TEL-AML
leuk_dist <- dist(leukemia_subset)
```

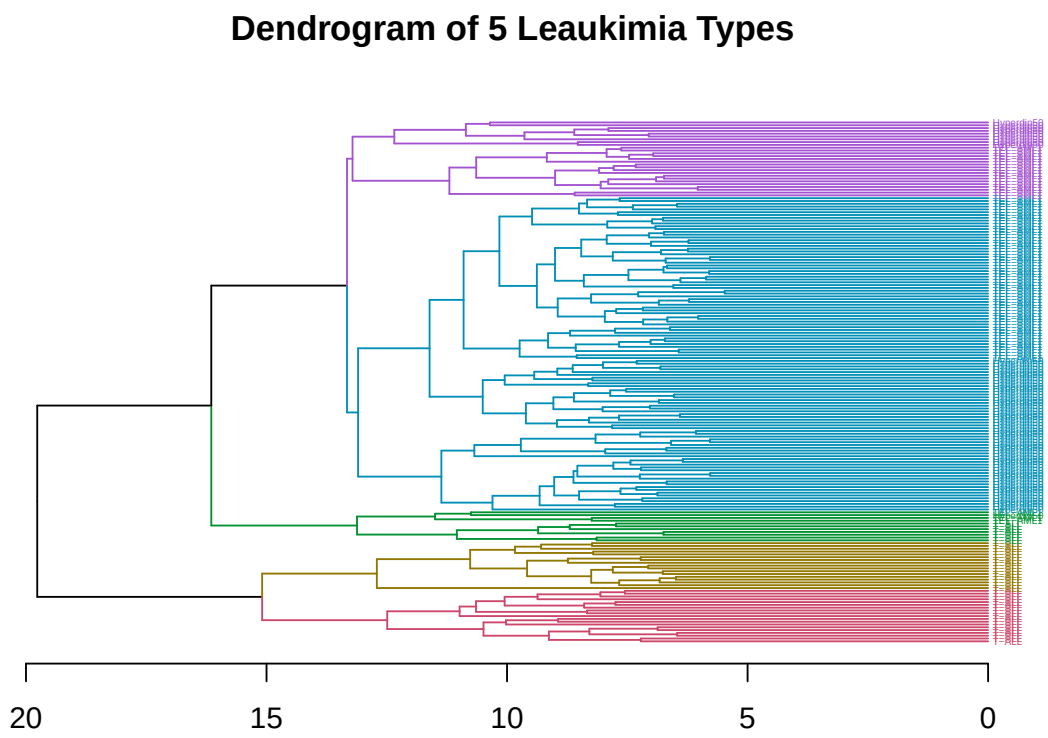
```
## Warning in dist(leukemia_subset): NAs introduced by coercion
```

```
leuk_hclust <- hclust(leuk_dist)

#First plot
dend1 = as.dendrogram(leuk_hclust)
dend1_cb = color_branches(dend1, k = 3)
dend1 = color_labels(dend1_cb, k = 3)
dend1 = set(dend1, "labels_cex", 0.3)
dend1 = set_labels(dend1, labels=leukemia_subset$Type[order.dendrogram(dend1)])
plot(dend1,horiz = TRUE, main = "Dendrogram of 3 Leaukimia Types")
```




```
#second plot
dend2 = as.dendrogram(leuk_hclust)
dend2_cb = color_branches(dend2, k = 5)
dend2 = color_labels(dend2_cb, k = 5)
#par(cex = 0.3)
dend2 = set(dend2, "labels_cex", 0.3)
dend2 = set_labels(dend2, labels=leukemia_subset$Type[order.dendrogram(dend2)])
plot(dend2,horiz = TRUE, main = "Dendrogram of 5 Leaukimia Types")
```


Dendrogram of 5 Leaukimia Types
20
15
10
5
0