Homework 3 - 131

Kalvin Goode (9454554), Ray Fan(8783920)

May 26, 2019

For this homework you will need use the following packages.

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

```
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. Logistic regression for drug use prediction

This problem has 3 parts for 131 students and 4 parts for 231 students. As mentioned, the data uses some strange encodings for variables. For instance, you may notice that the gender variable has type `double`. Here the value -0.48246 means male and 0.48246 means female. Age was recorded at a set of categories but rescaled to a mean 0 numeric variable (we will leave that variable as is). Similarly education is a scaled numeric quantity (we will also leave this variable as is). We will however, start by transforming gender, ethnicity, and country to factors, and the drug response variables as ordered factors:

```
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). Define a new factor response variable `recent_cannabis_use` which is "Yes" if a person has used cannabis within a year, and "No" otherwise. This can be done by checking if the `Cannabis` variable is *greater than or equal* to `CL3`. Hint: use `mutate` with the `ifelse` command. When creating the new factor set `levels` argument to `levels=c("No", "Yes")` (in that order).

```
drug_use=drug_use%>%
    mutate(recent_cannabis_use=factor(ifelse(Cannabis%in%c("CL3","CL4","CL5","CL6"),"Yes","No"),levels=c
drug_use
```

```
## # A tibble: 1,885 x 33
##       ID     Age Gender Education Country Ethnicity Nscore   Escore  Oscore
```

```
##    <int>   <dbl> <fct>      <dbl> <fct>   <fct>      <dbl>    <dbl>   <dbl>
##  1     1  0.498  Female   -0.0592 USA     Mixed:Wh~  0.313 -0.575   -0.583
##  2     2 -0.0785 Male      1.98   USA     White     -0.678  1.94     1.44
##  3     3  0.498  Male     -0.0592 USA     White     -0.467  0.805   -0.847
##  4     4 -0.952  Female    1.16   USA     White     -0.149 -0.806   -0.0193
##  5     5  0.498  Female    1.98   USA     White      0.735 -1.63    -0.452
##  6     6  2.59   Female   -1.23   UK      White     -0.678 -0.300   -1.56
##  7     7  1.09   Male      1.16   Austra~ White     -0.467 -1.09    -0.452
##  8     8  0.498  Male     -1.74   USA     White     -1.33   1.94    -0.847
##  9     9  0.498  Female   -0.0592 UK      White      0.630  2.57    -0.976
## 10    10  1.82   Male      1.16   USA     White     -0.246  0.00332 -1.42
## # ... with 1,875 more rows, and 24 more variables: Ascore <dbl>,
## #   Cscore <dbl>, Impulsive <dbl>, SS <dbl>, Alcohol <ord>, Amphet <ord>,
## #   Amyl <ord>, Benzos <ord>, Caff <ord>, Cannabis <ord>, Choc <ord>,
## #   Coke <ord>, Crack <ord>, Ecstasy <ord>, Heroin <ord>, Ketamine <ord>,
## #   Legalh <ord>, LSD <ord>, Meth <ord>, Mushrooms <ord>, Nicotine <ord>,
## #   Semer <ord>, VSA <ord>, recent_cannabis_use <fct>
```

(b). We will create a new tibble that includes a subset of the original variables. We will focus on all variables between `age` and `SS` as well as the new factor related to recent cannabis use. Create `drug_use_subset` with the command:

```
set.seed(25252)
drug_use_subset<-drug_use%>%select(Age:SS, recent_cannabis_use)

index=sample(nrow(drug_use_subset),1500)
drug_use_train=drug_use_subset[index,]
drug_use_test=drug_use_subset[-index,]

dim(drug_use_train)
```

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

```
dim(drug_use_test)
```

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

(c). Fit a logistic regression to model `recent_cannabis_use` as a function of all other predictors in `drug_use_train`. Fit this regression using the training data only. Display the results by calling the `summary` function on the logistic regression object.

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

```
##
## Call:
## glm(formula = recent_cannabis_use ~ ., family = "binomial", data = drug_use_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9083  -0.5788   0.1481   0.5322   2.6810
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.36768    0.77975   0.472  0.63726
## Age                        -0.91394    0.09414  -9.709  < 2e-16 ***
## GenderFemale               -0.71357    0.15919  -4.483 7.38e-06 ***
```

```
## Education                   -0.36788    0.08068  -4.560 5.12e-06 ***
## CountryCanada               12.47575  471.40287   0.026  0.97889
## CountryNew Zealand          -1.07682    0.33092  -3.254  0.00114 **
## CountryOther                -0.46641    0.48516  -0.961  0.33637
## CountryIreland              -0.17925    0.75160  -0.238  0.81150
## CountryUK                   -0.40715    0.37379  -1.089  0.27604
## CountryUSA                  -1.71557    0.19330  -8.875  < 2e-16 ***
## EthnicityAsian              -2.46471    1.43364  -1.719  0.08558 .
## EthnicityWhite               1.42331    0.77455   1.838  0.06612 .
## EthnicityMixed:White/Black   0.63408    1.17893   0.538  0.59068
## EthnicityOther               1.62124    0.90539   1.791  0.07335 .
## EthnicityMixed:White/Asian   1.92183    1.10881   1.733  0.08305 .
## EthnicityMixed:Black/Asian  13.95338  461.36516   0.030  0.97587
## Nscore                      -0.18350    0.09163  -2.003  0.04522 *
## Escore                      -0.21980    0.09738  -2.257  0.02399 *
## Oscore                       0.60433    0.09210   6.562 5.32e-11 ***
## Ascore                       0.06938    0.08191   0.847  0.39698
## Cscore                      -0.43361    0.09369  -4.628 3.69e-06 ***
## Impulsive                   -0.09466    0.10195  -0.929  0.35313
## SS                           0.65249    0.11222   5.814 6.08e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2074.0  on 1499  degrees of freedom
## Residual deviance: 1168.4  on 1477  degrees of freedom
## AIC: 1214.4
##
## Number of Fisher Scoring iterations: 13
```

2. Decision tree models of drug use

This problem has 3 parts for all students.

Construct a decision tree to predict `recent_cannabis_use` using all other predictors in `drug_use_train`. Set the value of the argument `control = tree_parameters` where `tree_parameters` are:

```
tree_parameters = tree.control(nobs=nrow(drug_use_train), minsize=10, mindev=1e-3)
tree_train=tree(recent_cannabis_use~.,data=drug_use_train,control=tree_parameters)
summary(tree_train)
```

```
##
## Classification tree:
## tree(formula = recent_cannabis_use ~ ., data = drug_use_train,
##     control = tree_parameters)
## Variables actually used in tree construction:
##  [1] "Country"   "SS"        "Age"       "Gender"    "Oscore"
##  [6] "Education" "Cscore"    "Nscore"    "Ascore"    "Escore"
## [11] "Impulsive"
## Number of terminal nodes:  131
## Residual mean deviance:  0.4244 = 581 / 1369
## Misclassification error rate: 0.09733 = 146 / 1500
```

This sets the smallest number of allowed observations in each leaf node to 10 and requires a deviance of at least 1e-3 to split a node.

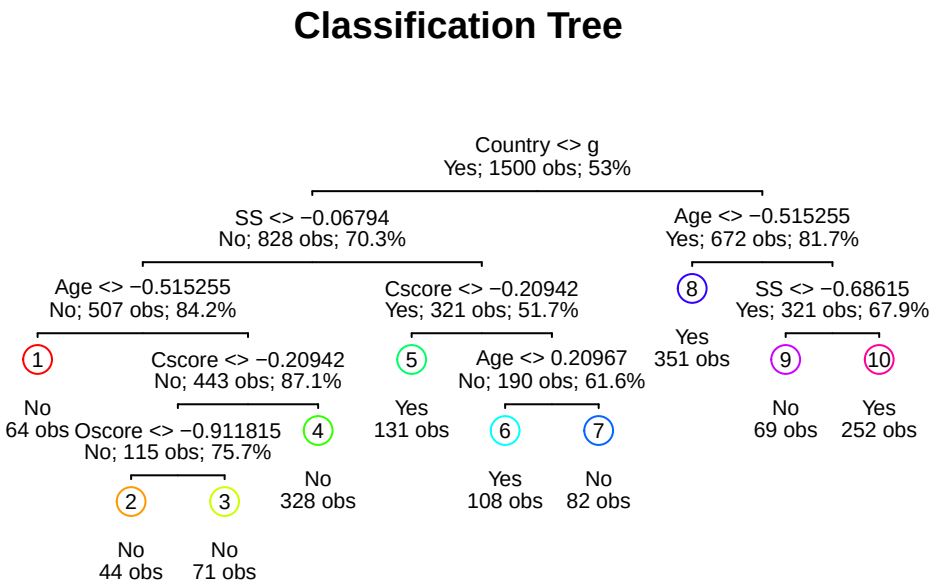
(a). Use 10-fold CV to select the a tree which minimizes the cross-validation misclassification rate. Use the function `cv.tree`, and set the argument `FUN=prune.misclass`. Note: you do not need to use a `do.chunk` function since the `tree` package will do cross validation for you. Find the size of the tree which minimizes the cross validation error. If multiple trees have the same minimum cross validated misclassification rate, set `best_size` to the smallest tree size with that minimum rate.

```
set.seed(1)
fold=10
tree=cv.tree(tree_train,K=fold,FUN=prune.misclass)
size=min(tree$size[tree$dev==min(tree$dev)])
size
```

```
## [1] 9
```

(b). Prune the tree to the size found in the previous part and plot the tree using the `draw.tree` function from the `maptree` package. Set `nodeinfo=TRUE`. Which variable is split first in this decision tree?

```
tree.prun=prune(tree_train,best=size)
draw.tree(tree.prun,nodeinfo=TRUE,size=2,cex=0.7)
title("Classification Tree")
```



(c). Compute and print the confusion matrix for the *test* data using the function `table(truth, predictions)` where `truth` and `predictions` are the true classes and the predicted classes from the tree model respectively. Note: when generated the predicted classes for the test data, set `type="class"` in the `predict` function. Calculate the true positive rate (TPR) and false positive rate (FPR) for the confusion matrix. Show how you arrived at your answer.

```
pred=predict(tree.prun,drug_use_test,type="class")
t=table(Truth=drug_use_test$recent_cannabis_use,Prediction=pred)
```

```
t
```

```
##      Prediction
## Truth  No Yes
##   No  135  46
##   Yes  31 173
```

```
FPR=t[2,2]/(t[2,2]+t[2,1])
TPR=t[1,2]/(t[1,1]+t[1,2])
```

```
FPR
```

```
## [1] 0.8480392
```

```
TPR
```

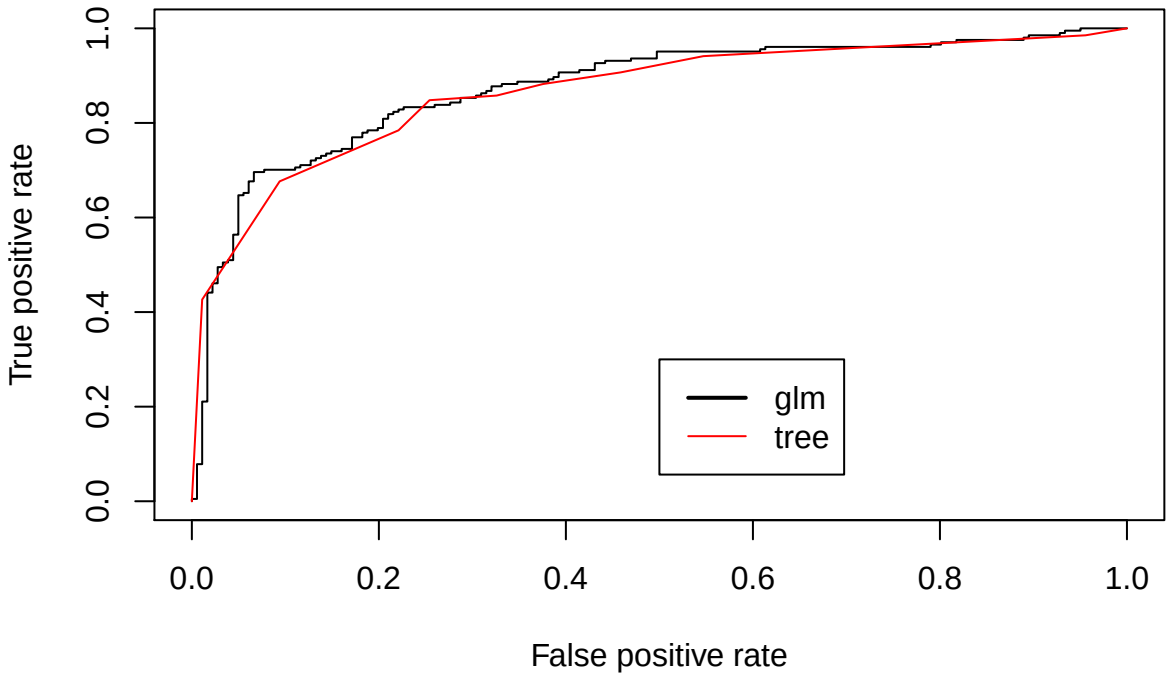
```
## [1] 0.2541436
```

3. Model Comparison

This problem has 2 parts for all students.

(a). Plot the ROC curves for both the logistic regression fit and the decision tree on the same plot. Use `drug_use_test` to compute the ROC curves for both the logistic regression model and the best pruned tree model.

```
pred2=predict(fit,newdata=drug_use_test,type="response")
prediction1=prediction(pred2,drug_use_test$recent_cannabis_use)
performance1=performance(prediction1,measure="tpr",x.measure="fpr")
plot(performance1)
treepred=predict(tree.prun,drug_use_test,type="vector")[,2]
prediction2=prediction(treepred,drug_use_test$recent_cannabis_use)
performance2=performance(prediction2,measure="tpr",x.measure="fpr")
plot(performance2,col='red',add=TRUE)
legend(0.5,0.3,col=c(1,2),legend=c("glm","tree"),lty=1,lwd=c(2,1))
```



(b). Compute the AUC for both models and print them. Which model has larger AUC?

```
aucp1=performance(prediction1,measure="auc")
aucp2=performance(prediction2,measure="auc")
aucp1@y.values
```

```
## [[1]]
## [1] 0.8747969
```

```
aucp2@y.values
```

```
## [[1]]
## [1] 0.8659544
```

According to the result, the logistic model has a larger AUC.

4. Clustering and dimension reduction for gene expression data

This problem involves the analysis of gene expression data from 327 subjects from Yeoh *et al* (2002). The data set includes abundance levels for 3141 genes and a class label indicating one of 7 leukemia subtypes the patient was diagnosed with. The paper describing their analysis of this data can be found here. Read in the csv data in `leukemia_data.csv`. It is posted on Piazza in the resources tab with the homework:

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

(a). The class of the first column of `leukemia_data`, `Type`, is set to `character` by default. Convert the `Type` column to a factor using the `mutate` function. Use the `table` command to print the number of patients with each leukemia subtype. Which leukemia subtype occurs the least in this data?

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

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

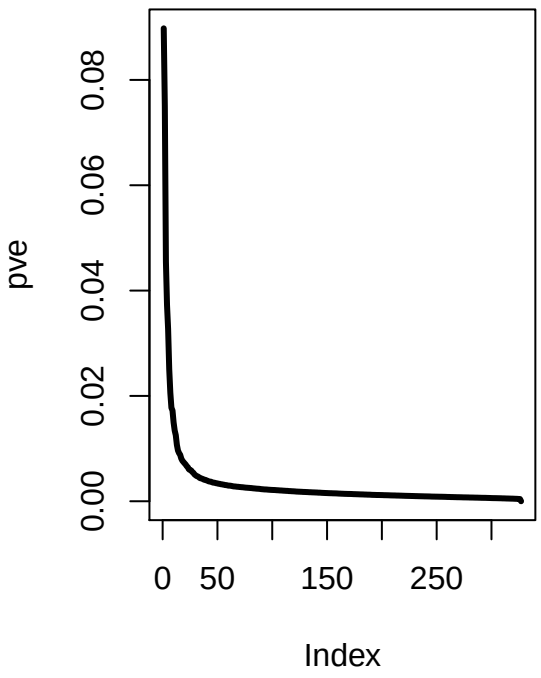
```
leukemia_data$Type[which.min(leukemia_data$Type)]
```

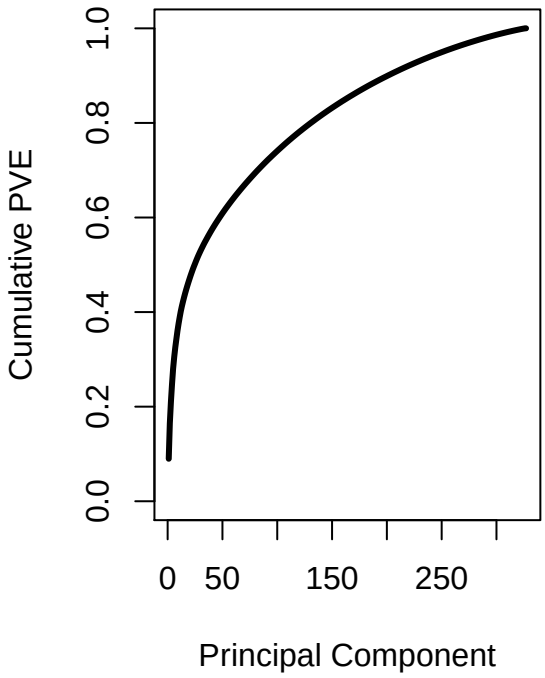
```
## [1] BCR-ABL
## Levels: BCR-ABL E2A-PBX1 Hyperdip50 MLL OTHERS T-ALL TEL-AML1
```

According to the result above, leukemia subtype occurs the least in this data is BCR-ABL.

(b). Run PCA on the leukemia data using `prcomp` function with `scale=TRUE` and `center=TRUE` (this scales each gene to have mean 0 and variance 1). Make sure you exclude the `Type` column when you run the PCA function (we are only interested in reducing the dimension of the gene expression values and PCA doesn't work with categorical data anyway). Plot the proportion of variance explained by each principal component (PVE) and the cumulative PVE side-by-side.

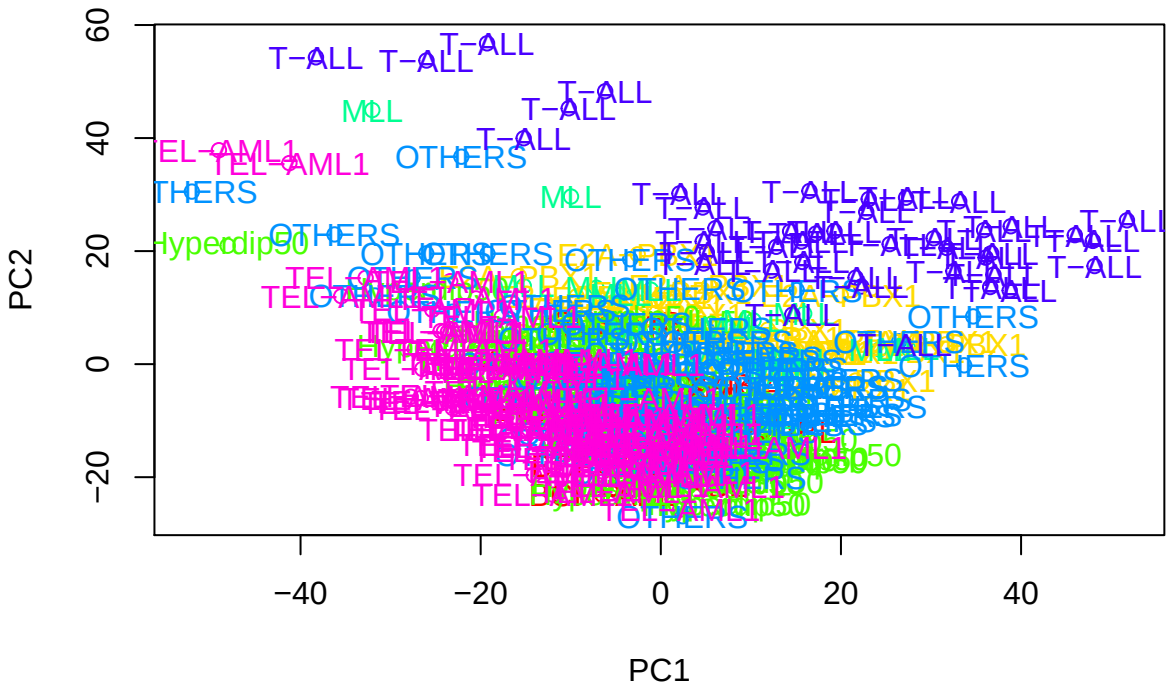
```
leukemia_new <- leukemia_data %>% select(-1)
pca<-leukemia_new %>%
prcomp(scale=TRUE,center=TRUE)
sdev<-pca$sdev
pve <- sdev^2 / sum(sdev^2)
cumulative_pve <- cumsum(pve)

## 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,
     xlab = "Principal Component",
     ylab = "Cumulative PVE",ylim = c(0,1))
```





(c). Use the results of PCA to project the data into the first two principal component dimensions. `prcomp` returns this dimension reduced data in the first columns of `x`. Plot the data as a scatter plot using `plot` function with `col=plot_colors` where `plot_colors` is defined

```
rainbow_colors <- rainbow(7)
plot_colors <- rainbow_colors[leukemia_data$Type]
plot(pca$x,col=plot_colors)
text(pca$x,labels=leukemia_data$Type,col = plot_colors)
```



Due to the unclearness of this picture, a text result is returned below:

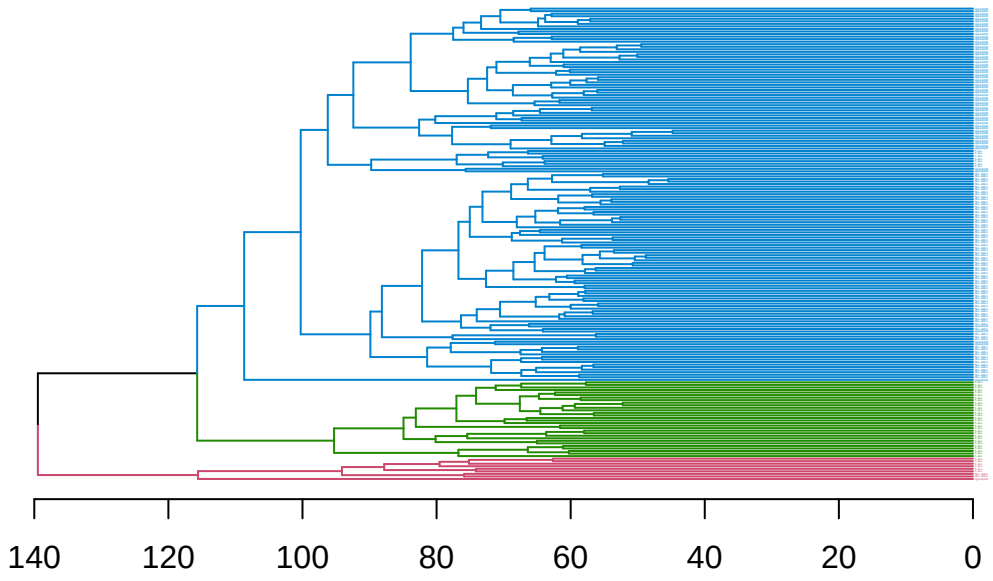
```
pc1<-pca$rotation[,1]
sort.pc1<-sort(abs(pc1),decreasing = T)
head(sort.pc1)
```

```
##     SEMA3F       CCT2       LDHB      COX6C     SNRPD2       ELK3
## 0.04517148 0.04323818 0.04231619 0.04183480 0.04179822 0.04155821
```

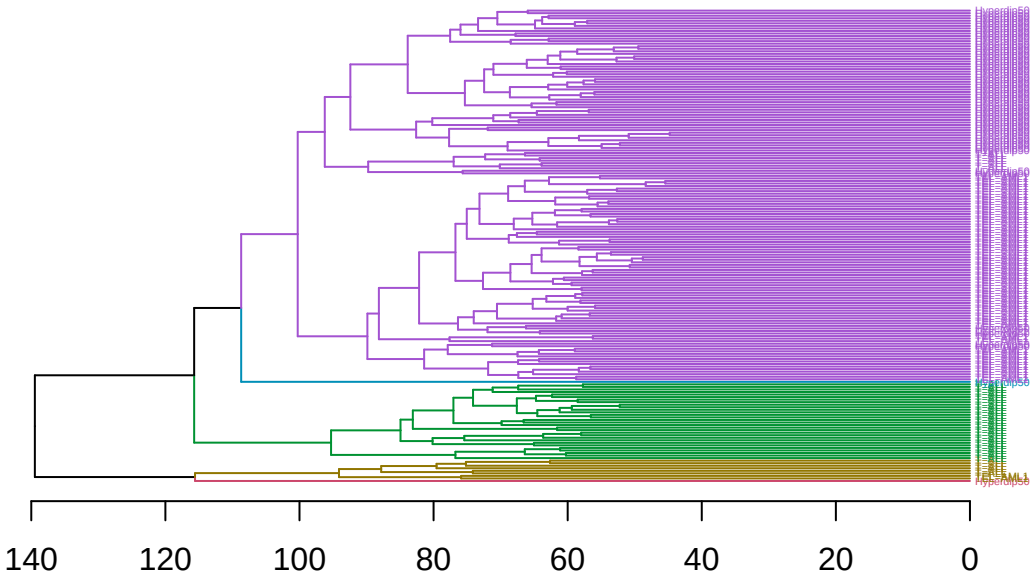
Therefore, genes with highest absolute loadings for PC1 are SEMA3F,CCT2,LDHB,COX6C,SNRPD2 and ELK3.

(f).

```
leusub=leukemia_data%>%
      filter(leukemia_data$Type %in% c("T-ALL","TEL-AML1","Hyperdip50"))

dist=dist(scale(leusub[,-1]),method="euclidean")
leuhier=hclust(dist,method="complete")


dend1=as.dendrogram(leuhier) %>%
                color_branches(k=3) %>%
                color_labels(k=3) %>%
                set("labels_cex",0.1) %>%
            set_labels(.,labels=leusub$Type[order.dendrogram(.)])%>%
            plot(horiz=TRUE)
```




```
dend2=as.dendrogram(leuhier) %>%
                color_branches(k=5) %>%
                color_labels(k=5) %>%
                set("labels_cex",0.3) %>%
            set_labels(.,labels=leusub$Type[order.dendrogram(.)])%>%
            plot(horiz=TRUE)
```


Hyperdiploid50
ETV6-RUNX1
Hyperdiploid50
Hyperdiploid50
Hyperdiploid50
Hyperdiploid50
Hyperdiploid50
TCF3-PBX1
Hyperdiploid50
140
120
100
80
60
40
20
0