PSTAT 131/231 HW #3

Lash Tan (231) and Jacobo Pereira-Pacheco (131)

5/20/2018

```
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

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

```
library(tree)
library(maptree)
```

```
## Loading required package: cluster
```

```
## Loading required package: rpart
```

```
library(class)
library(lattice)
library(dplyr)
library(ggridges)
library(lattice)
```

```
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'))
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   ID = col_integer(),
##   Age = col_double(),
##   Gender = col_double(),
##   Education = col_double(),
```

```
##   Country = col_double(),
##   Ethnicity = col_double(),
##   Nscore = col_double(),
##   Escore = col_double(),
##   Oscore = col_double(),
##   Ascore = col_double(),
##   Cscore = col_double(),
##   Impulsive = col_double(),
##   SS = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

Question 1

Logistic regression for drug use prediction

```
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")))
```

b)

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

```
set.seed(1)
train.indeces = sample(1:nrow(drug_use_subset), 1500)
drug_use_train = drug_use_subset[train.indeces,]
cat("Dimensions of drug_use_train:", dim(drug_use_train))
```

```
## Dimensions of drug_use_train: 1500 13
```

```
drug_use_test = drug_use_subset[-train.indeces,]
cat("Dimensions of drug_use_test:", dim(drug_use_test))
```

```
## Dimensions of drug_use_test: 385 13
```

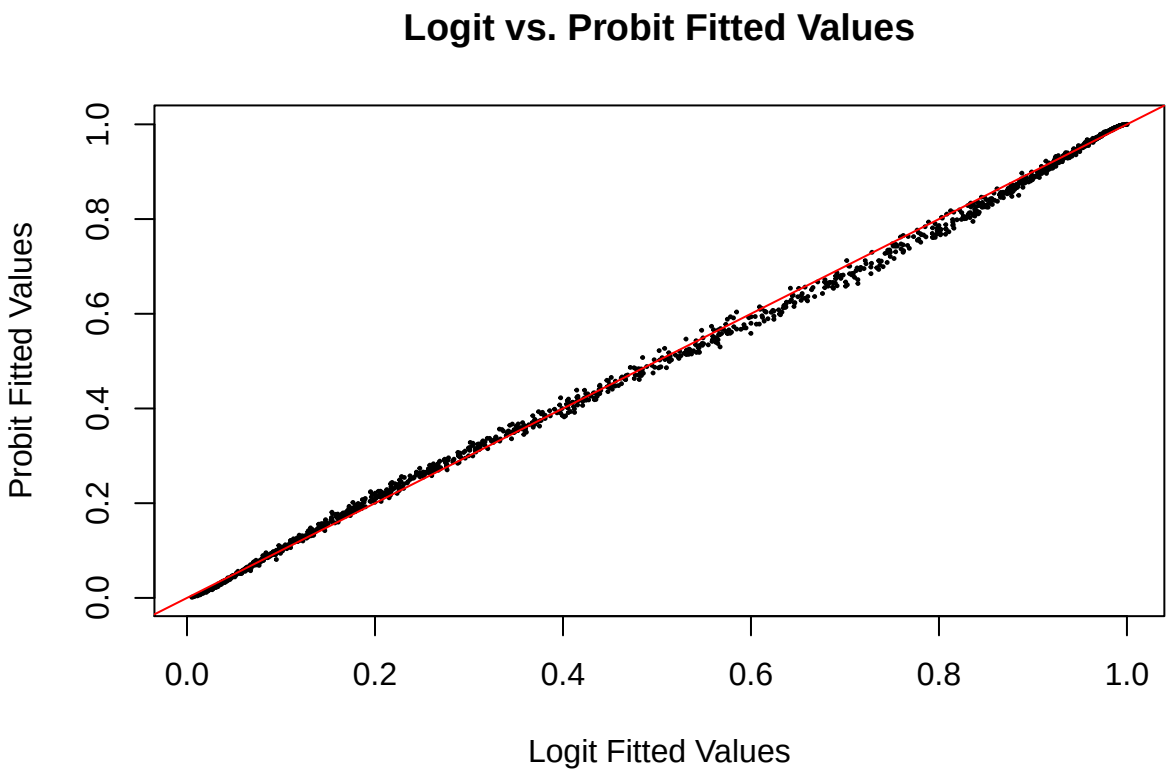
c)

```
drug.fit.logit <- glm(recent_cannabis_use ~ ., family = "binomial", data = drug_use_train) ## default li
summary(drug.fit.logit)
```

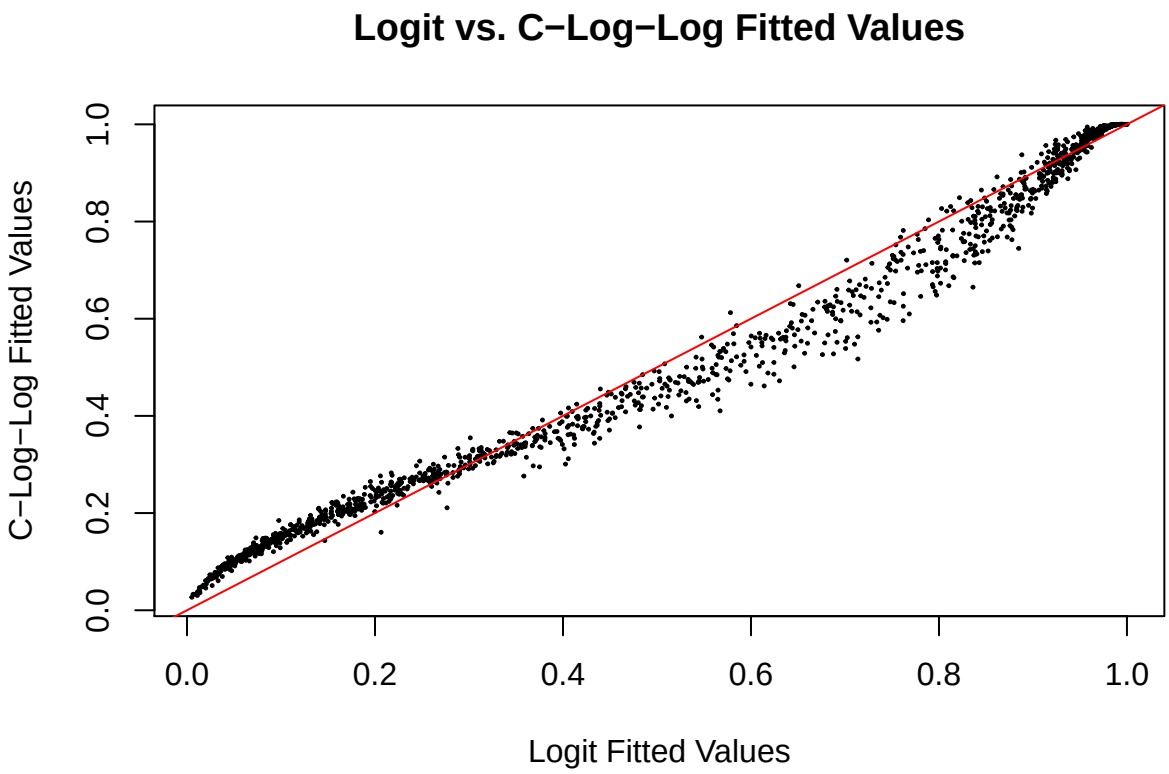
```
##
## Call:
## glm(formula = recent_cannabis_use ~ ., family = "binomial", data = drug_use_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0024  -0.5996   0.1512   0.5410   2.7525
##
## Coefficients:
##                             Estimate Std. Error z value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

d)

```
drug.fit.probit <- glm(recent_cannabis_use ~ ., family = binomial(link = "probit"), data = drug_use_tra
plot(drug.fit.logit$fitted.values, drug.fit.probit$fitted.values,
     xlab = 'Logit Fitted Values', ylab = 'Probit Fitted Values',
     main = 'Logit vs. Probit Fitted Values', pch=19, cex=0.2)
abline(a=0, b=1, col="red")
```




```
drug.fit.cloglog <- glm(recent_cannabis_use ~ ., family = binomial(link = "cloglog"), data = drug_use_t
plot(drug.fit.logit$fitted.values, drug.fit.cloglog$fitted.values,
     xlab = 'Logit Fitted Values', ylab = 'C-Log-Log Fitted Values',
     main = 'Logit vs. C-Log-Log Fitted Values', pch=19, cex=0.2)
abline(a=0, b=1, col="red")
```



Based on the two plots of fitted values, the *probit* fitted values more closely resemble the fitted values for *logit*. There only seems to be a slight overestimate for the first half quantile and a slight underestimate for the second half quantile of the *probit* fitted values in comparison with the *logit* fitted values. The *cloglog* fitted values seem to overestimate the tails and more significantly underestimate the rest of the data, also in comparison with the *logit* fitted values. Also, the *probit* and *logit* fitted values seem to very closely predict similar trends in probabilities. That is, while there are minor discrepancies between the estimates using these two link functions, there is very little variation between these differences. The *cloglog* fitted values, on the other hand, have very wide variation in comparison, especially toward the median of these values. This variation is shown by the amount of spread between points on the plots above.

Question 2

Decision tree models of drug use

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

a)

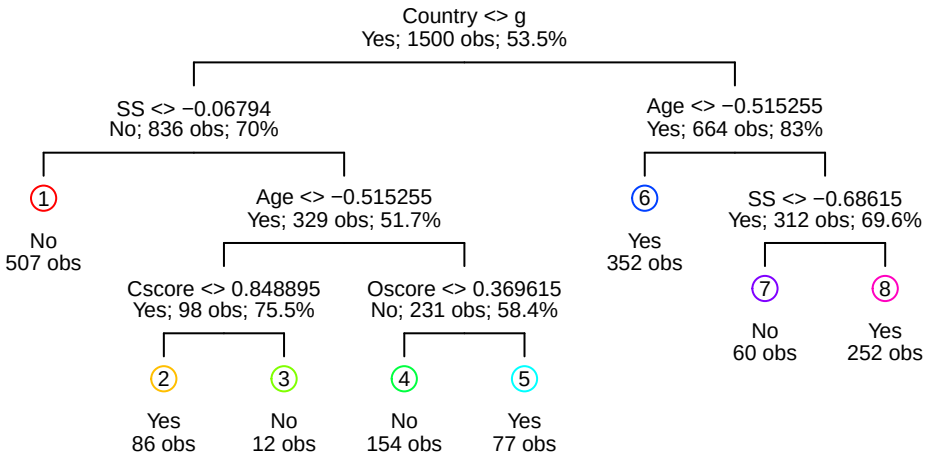
```
set.seed(1)
drug.tree.cv <- cv.tree(drug.tree, FUN = prune.misclass, K = 10)
```

```
(best_size <- which.min(rev(drug.tree.cv$dev)) %>% ## reverses the list of deviances to find first mini
              rev(drug.tree.cv$size)[.]) ## chooses the element associated with the correct index found
```

```
## [1] 8
```

b)

```
drug.tree.prune <- prune.tree(drug.tree, best = best_size , method = "misclass")
draw.tree(drug.tree.prune, nodeinfo = T, cex = .7)
```



We can see that the first split of our tree is by the `Country` variable.

c)

```
predict.tree <- predict(drug.tree.prune, drug_use_test, type = 'class')
(conf.mat <- table(predict.tree, drug_use_test$recent_cannabis_use, dnn = c("Prediction", "Truth")))
```

```
##           Truth
## Prediction  No Yes
##        No  152  42
##        Yes  36 155
```

```
tpr <- conf.mat[2,2] / sum(conf.mat[,2])
fpr <- conf.mat[2,1] / sum(conf.mat[,1])

cat("The TPR of our predictions is", tpr, "and the FPR is", fpr)
```

```
## The TPR of our predictions is 0.786802 and the FPR is 0.1914894
```

As the *true positive rate* (TPR) is calculated by $\frac{TP}{TP+FN}$, we divide the bottom right element by the second column of our confusion matrix. Likewise, the *false positive rate* (FPR) is calculated by $\frac{FP}{FP+TN}$ which can

be obtained by dividing the lower left element by the first column of our confusion matrix.

Question 3

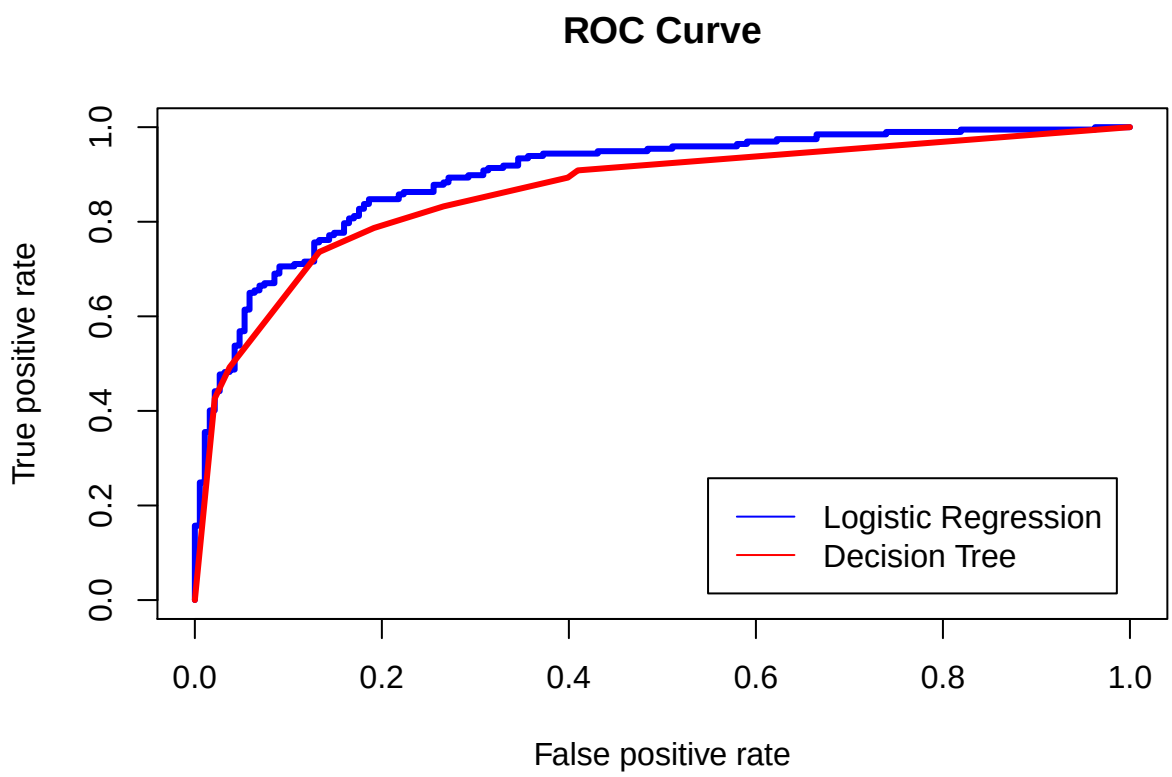
Model Comparison

a)

```
predict.logistic <- predict(drug.fit.logit, drug_use_test, type = 'response')
prediction.logistic <- prediction(predict.logistic, drug_use_test$recent_cannabis_use)
perf.logistic <- performance(prediction.logistic, measure = "tpr", x.measure = "fpr")

prob.tree <- predict(drug.tree.prune, drug_use_test, type="vector")
prediction.tree <- prediction(prob.tree[,2], drug_use_test$recent_cannabis_use)
perf.tree <- performance(prediction.tree, measure = "tpr", x.measure = "fpr")
```

```
plot(perf.logistic, col='blue', lwd=3, main="ROC Curve")
plot(perf.tree, col='red', lwd=3, main="ROC Curve", add="T")
legend("bottomright", inset = .05, legend=c("Logistic Regression", "Decision Tree"),
       col=c("blue", "red"), lty=1, cex=1)
```




```
auc.logistic <- performance(prediction.logistic,"auc")@y.values[[1]]
auc.tree <- performance(prediction.tree,"auc")@y.values[[1]]
cat("AUC for logistic regression:", auc.logistic)
```

```
## AUC for logistic regression: 0.8973971
```

```r
cat("AUC for decision tree:", auc.tree)
```

```
## AUC for decision tree: 0.8633087
```

So, logistic regression generally gives us the better model as it has a higher AUC value.

Question 4

Clustering and dimension reduction for gene expression data

```r
# rm(list=ls()) ## environment variables up to here can be reset
leukemia_data <- read_csv("leukemia_data.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   Type = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

a)

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

table(leukemia_data$Type)
```

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

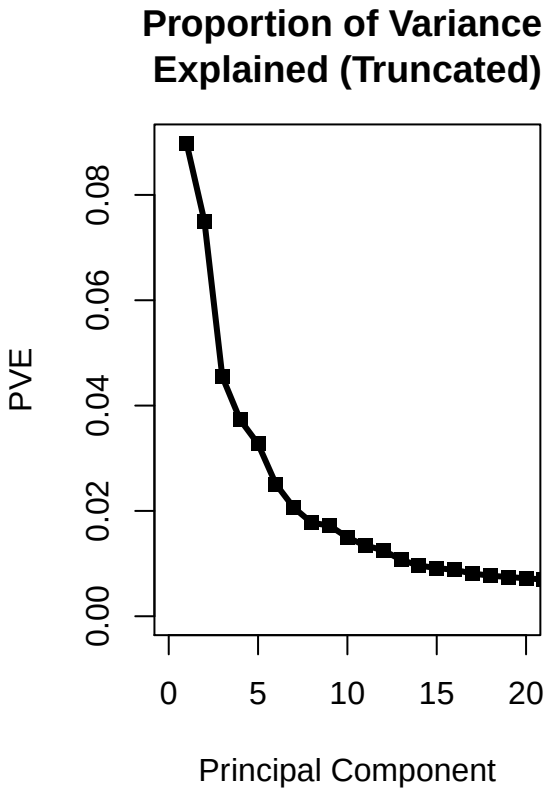
The BCR-ABL leukemia type appears the least in this dataset.

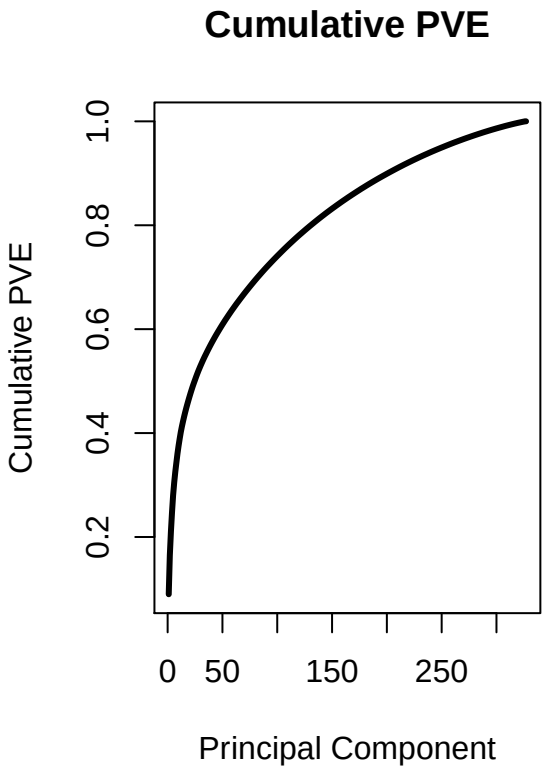
b)

```r
leuk.pr.out <- leukemia_data %>%
        select_at(vars(-Type)) %>%
        prcomp(., scale = TRUE, center = TRUE)
leuk.pr.var <- leuk.pr.out$sdev^2
pve <- leuk.pr.var/sum(leuk.pr.var)
cumulative_pve <- cumsum(pve)
```

```r
par(mfrow = c(1,2))

plot(pve, type="l", lwd=3, xlim = c(0,20),
     xlab = 'Principal Component', ylab = 'PVE', main = 'Proportion of Variance \nExplained (Truncated)')
points(pve, pch = 15)
plot(cumulative_pve, type="l", lwd=3, xlab = 'Principal Component',
     ylab = 'Cumulative PVE', main = 'Cumulative PVE')
```



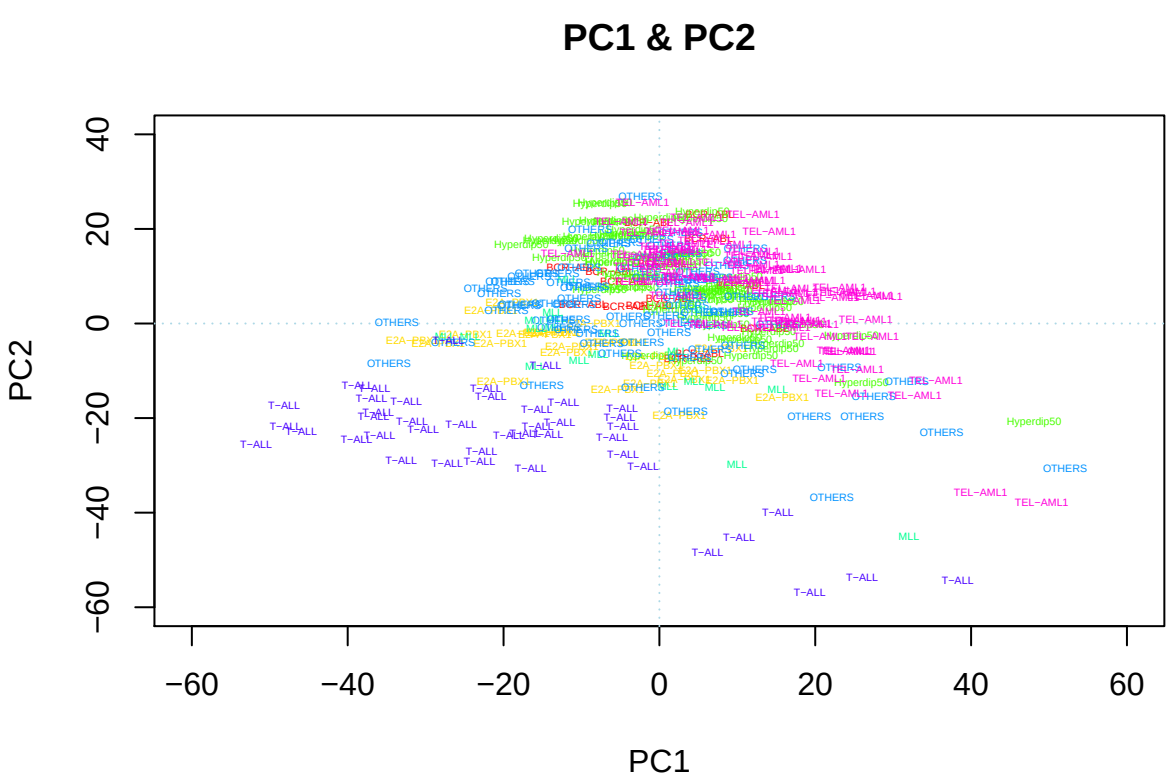


We have zoomed in on the PVE plot to better visualize the optimal number of principal components to choose from. It is clear that beyond 20 principal components, adding an additional principal component negligibly adds to the PVE.

c)

```
rainbow_colors <- rainbow(7)
plot_colors <- rainbow_colors[leukemia_data$Type]

new_coords <- leuk.pr.out$x[, 1:2]
plot(new_coords, xlim=c(-60, 60), ylim=c(-60, 40), cex=0, main = 'PC1 & PC2')
text(-new_coords, labels=leukemia_data$Type, cex=0.3, col = plot_colors)
abline(h=0, v=0, col="lightblue", lty=3)
```



The T-ALL group is the most separated by the rest of the other types along the PC1 axis. This group clearly contains the lowest values while extending all the way out toward the maximum values of PC1.

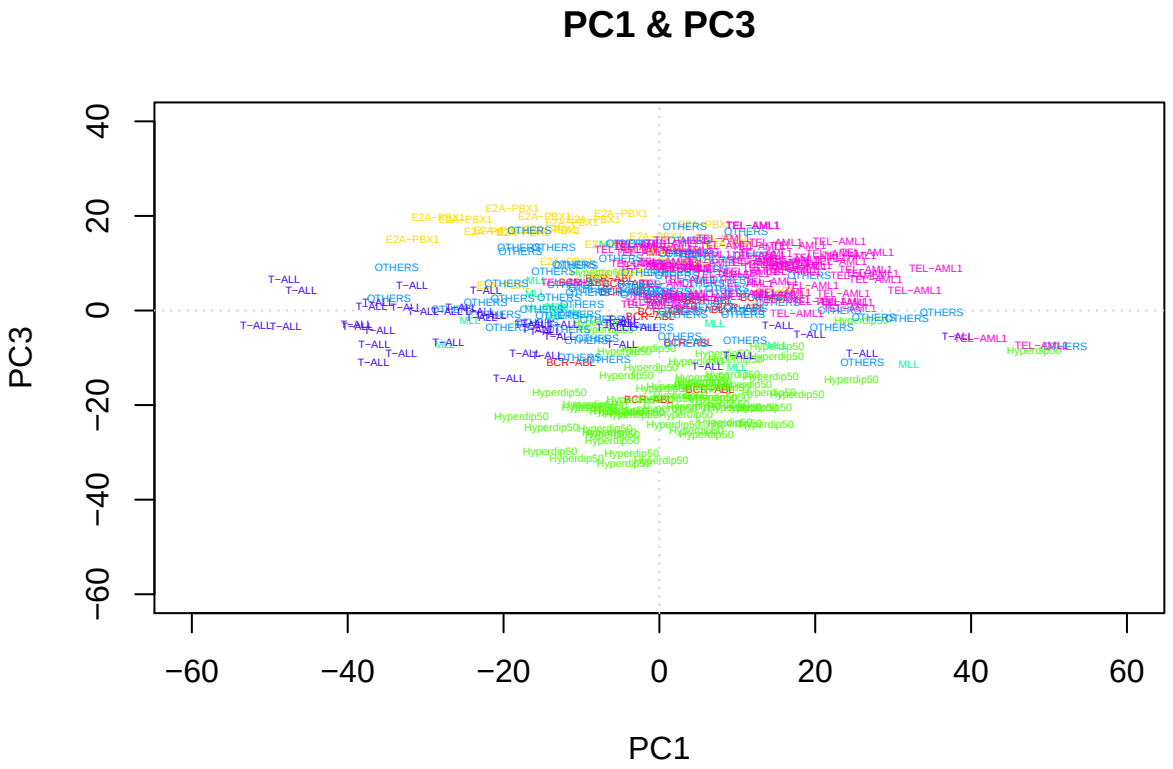
```
head(sort(abs(leuk.pr.out$rotation[, 1]), TRUE))
```

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

These 6 genes have the highest absolute loadings for PC1.

d)

```
new_new_coords <- leuk.pr.out$x[, c(1,3)]
plot(new_new_coords, xlim=c(-60, 60), ylim=c(-60, 40), cex=0, main = 'PC1 & PC3')
text(-new_new_coords, labels=leukemia_data$Type, cex=0.3, col = plot_colors)
abline(h=0, v=0, col="lightblue", lty=3)
```

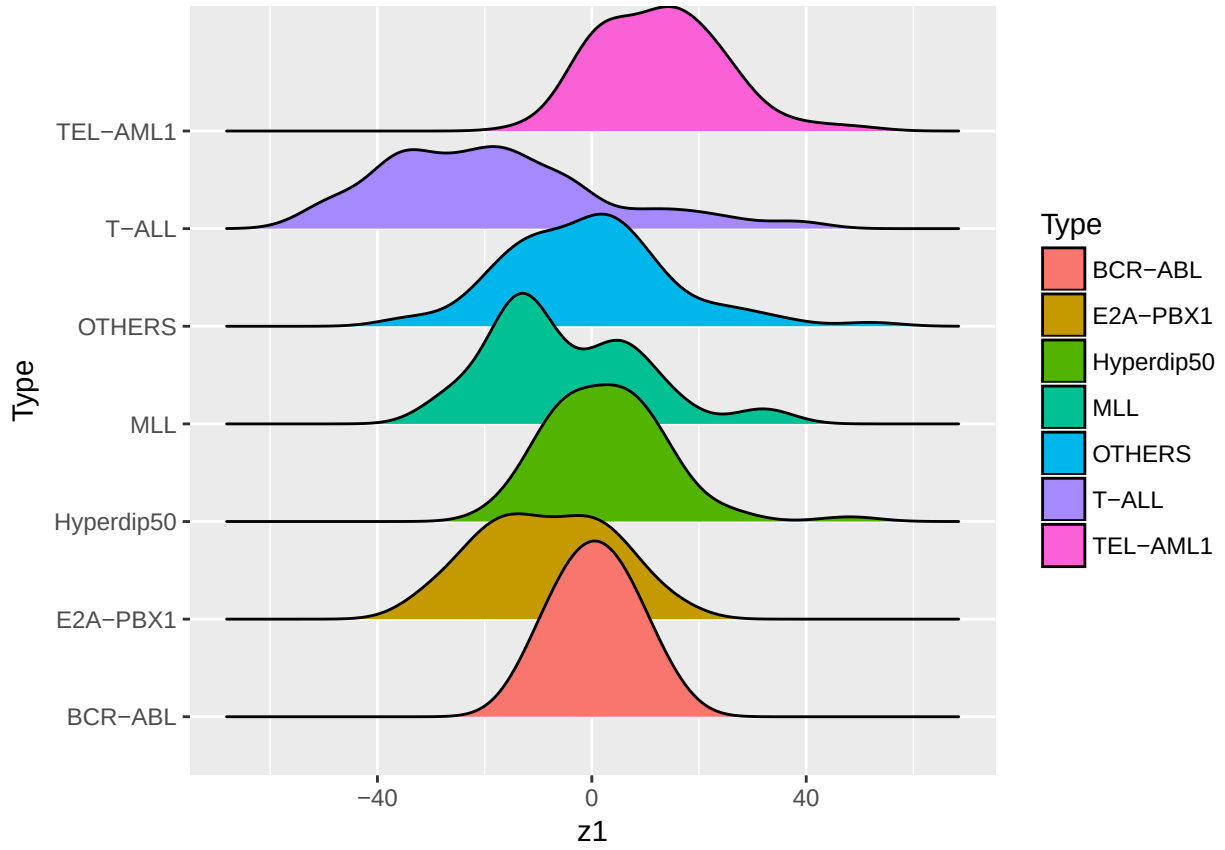


It's hard to tell if PC3 is better at discriminating between leukemia types from comparing these two graphs. There may be a bit more separation between these types for PC3, but it would also depend on which groups you want to separate.

e)

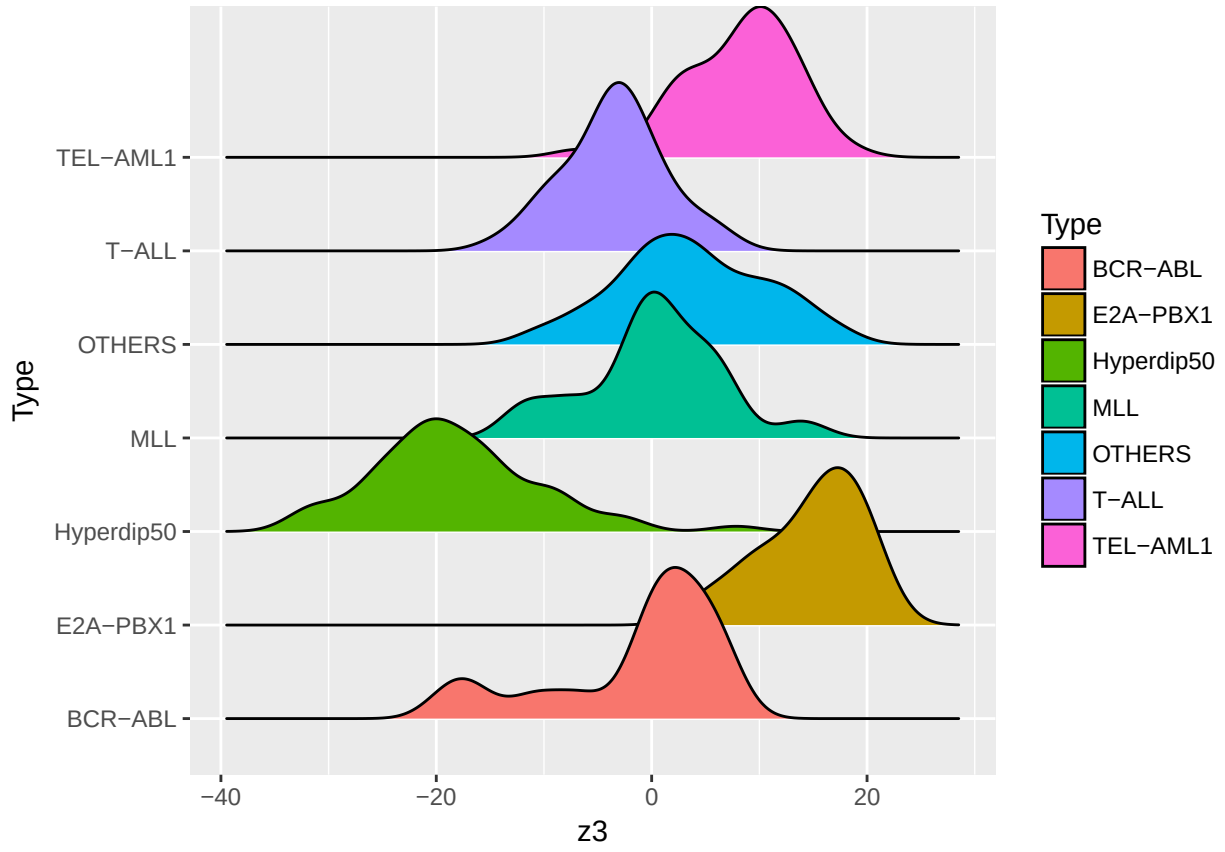
```
first.proj <- tibble(z1 = -leuk.pr.out$x[,1], Type = leukemia_data$Type)
ggplot(data = first.proj, mapping = aes(x = z1, y = Type, fill = Type)) +
  geom_density_ridges()
```

```
## Picking joint bandwidth of 5.46
```




```
third.proj <- tibble(z3 = -leuk.pr.out$x[,3], Type = leukemia_data$Type)
ggplot(data = third.proj, mapping = aes(x = z3, y = Type, fill = Type)) +
  geom_density_ridges()
```

```
## Picking joint bandwidth of 2.29
```

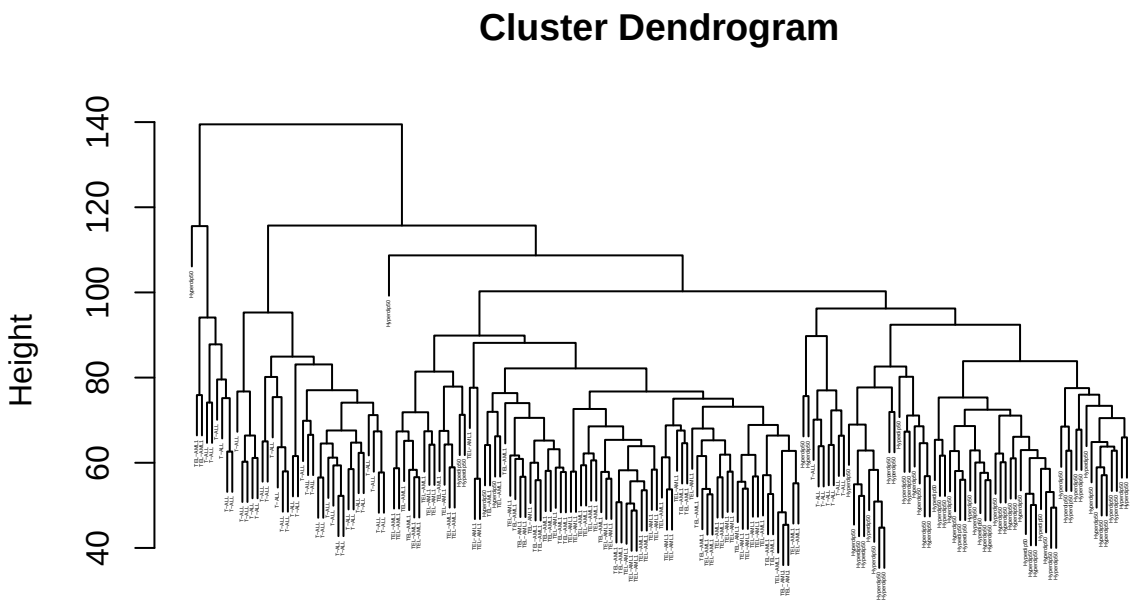


It appears that the *Hyperdip50* and *BCR-ABL* leukemia types are nearly indistinguishable when the gene expression data is projected onto the first PC direction, but they are very distinguishable when projecting onto the third PC direction.

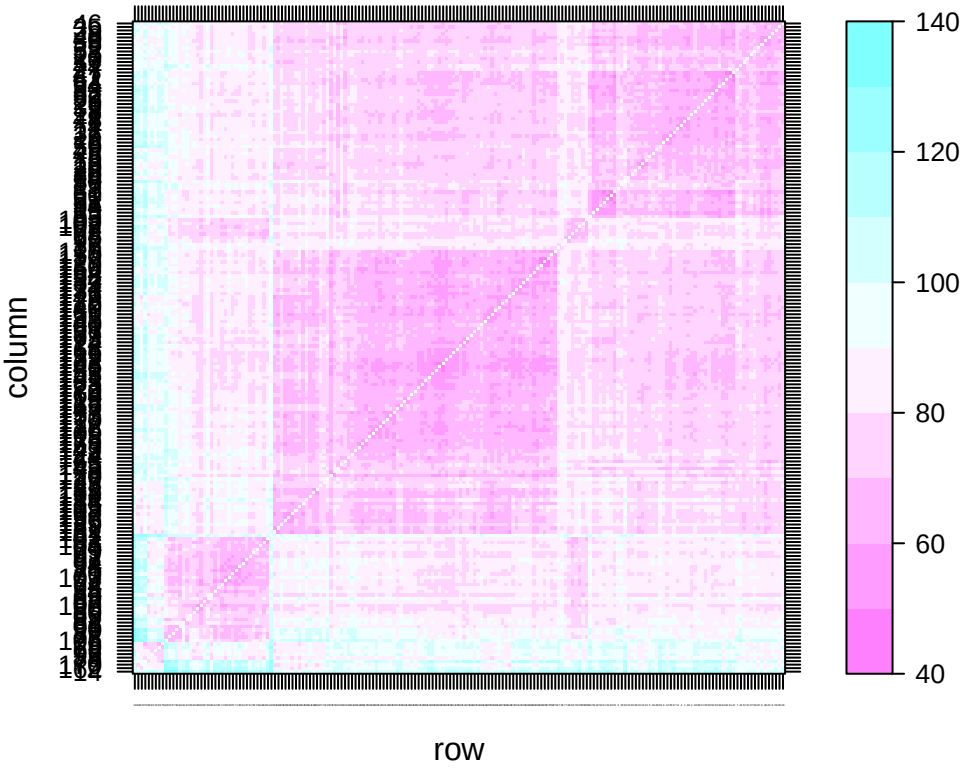
f)

```
leukemia_subset <- leukemia_data %>% filter(Type %in% c('T-ALL','TEL-AML1','Hyperdip50'))
leukemia_subset[,-1] <- scale(leukemia_subset[,-1], center = T, scale = T)
leuk.dist <- dist(leukemia_subset[,-1])

set.seed(1)
leuk.hclust <- hclust(leuk.dist)
plot(leuk.hclust, labels = leukemia_subset$Type, cex = .2)
```




```
levelplot(as.matrix(leuk.dist)[leuk.hclust$order, leuk.hclust$order], at=pretty(c(44.79, 139.5), n=10),
```




```
leukemia_subset %>% group_by(Type) %>% summarise(count = n())
```

```
## # A tibble: 3 x 2
##         Type count
##       <fctr> <int>
## 1 Hyperdip50    64
## 2      T-ALL    43
## 3   TEL-AML1    79
```

```
leuk.hclust$order
```

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

Based on the information above, the three blocks in the levelplot mainly represent Hyperdip50 in the bottom left, TEL-AML1 in the middle, and T-ALL in the upper right (with some overlapping). As pink represents shorter distances while blue represents larger distances, it seems reasonable to assume that TEL-AML1 and T-ALL are more similar to one another.