131 HW3

```
drug_use <- read_csv('drug.csv',
col_names = c('ID','Age','Gender','Education','Country','Ethnicity', 'Nscore','Escore','Oscore','Ascore
```

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

```
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")))
```

1(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.

```
drug_use <- drug_use %>% mutate(recent_cannabis_use = as.factor(ifelse(Cannabis < 'CL3', 'No', 'Yes')))
str(drug_use)
```

```
## spec_tbl_df[,33] [1,885 x 33] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ID                : num [1:1885] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Age               : num [1:1885] 0.4979 -0.0785 0.4979 -0.952 0.4979 ...
##  $ Gender            : Factor w/ 2 levels "Male","Female": 2 1 1 2 2 2 1 1 2 1 ...
##  $ Education         : num [1:1885] -0.0592 1.9844 -0.0592 1.1637 1.9844 ...
```

```
##  $ Country           : Factor w/ 7 levels "Australia","Canada",..: 7 7 7 7 7 6 1 7 6 7 ...
##  $ Ethnicity         : Factor w/ 7 levels "Black","Asian",..: 6 3 3 3 3 3 3 3 3 3 ...
##  $ Nscore            : num [1:1885] 0.313 -0.678 -0.467 -0.149 0.735 ...
##  $ Escore            : num [1:1885] -0.575 1.939 0.805 -0.806 -1.633 ...
##  $ Oscore            : num [1:1885] -0.5833 1.4353 -0.8473 -0.0193 -0.4517 ...
##  $ Ascore            : num [1:1885] -0.917 0.761 -1.621 0.59 -0.302 ...
##  $ Cscore            : num [1:1885] -0.00665 -0.14277 -1.0145 0.58489 1.30612 ...
##  $ Impulsive         : num [1:1885] -0.217 -0.711 -1.38 -1.38 -0.217 ...
##  $ SS                : num [1:1885] -1.181 -0.216 0.401 -1.181 -0.216 ...
##  $ Alcohol           : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 6 6 7 5 5 3 7 6 5 7 ...
##  $ Amphet            : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 3 3 1 1 2 1 1 1 1 2 ...
##  $ Amyl              : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 1 3 1 1 2 1 1 1 1 1 ...
##  $ Benzos            : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 3 1 1 4 1 1 1 1 1 2 ...
##  $ Caff              : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 7 7 7 6 7 7 7 7 7 7 ...
##  $ Cannabis          : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 1 5 4 3 4 1 2 1 1 2 ...
##  $ Choc              : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 6 7 5 5 7 5 6 5 7 7 ...
##  $ Coke              : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 1 4 1 3 1 1 1 1 1 1 ...
##  $ Crack             : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Ecstasy           : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 1 5 1 1 2 1 1 1 1 1 ...
##  $ Heroin            : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Ketamine          : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 1 3 1 3 1 1 1 1 1 1 ...
##  $ Legalh            : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 1 1 1 1 2 1 1 1 1 1 ...
##  $ LSD               : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 1 3 1 1 1 1 1 1 1 1 ...
##  $ Meth              : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 1 4 1 1 1 1 1 1 1 1 ...
##  $ Mushrooms         : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 1 1 2 1 3 1 1 1 1 1 ...
##  $ Nicotine          : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 3 5 1 3 3 7 7 1 7 7 ...
##  $ Semer             : Ord.factor w/ 5 levels "CL0"<"CL1"<"CL2"<..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ VSA               : Ord.factor w/ 7 levels "CL0"<"CL1"<"CL2"<..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ recent_cannabis_use: Factor w/ 2 levels "No","Yes": 1 2 2 1 2 1 1 1 1 1 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ID = col_double(),
##   ..   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(),
##   ..   Alcohol = col_character(),
##   ..   Amphet = col_character(),
##   ..   Amyl = col_character(),
##   ..   Benzos = col_character(),
##   ..   Caff = col_character(),
##   ..   Cannabis = col_character(),
##   ..   Choc = col_character(),
##   ..   Coke = col_character(),
##   ..   Crack = col_character(),
##   ..   Ecstasy = col_character(),
```

```
##   ..   Heroin = col_character(),
##   ..   Ketamine = col_character(),
##   ..   Legalh = col_character(),
##   ..   LSD = col_character(),
##   ..   Meth = col_character(),
##   ..   Mushrooms = col_character(),
##   ..   Nicotine = col_character(),
##   ..   Semer = col_character(),
##   ..   VSA = col_character()
##   .. )
```

1(b)Split drug_use_subset into a training data set and a test data set called drug_use_train and drug_use_test. The training data should include 1500 randomly sampled observation and the test data should include the remaining observations in drug_use_subset.

```
drug_use_subset <- drug_use %>% select(Age:SS, recent_cannabis_use)
train <- sample(1:nrow(drug_use_subset),1500)
drug_use_train <- drug_use_subset[train,]
drug_use_test <- drug_use_subset[-train,]
dim(drug_use_train)
```

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

```
dim(drug_use_test)
```

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

1(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.

```
log_mod <- glm(recent_cannabis_use~. , data = drug_use_train , family = binomial)
summary(log_mod)
```

```
##
## Call:
## glm(formula = recent_cannabis_use ~ ., family = binomial, data = drug_use_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7831  -0.5756   0.1507   0.5249   2.7480
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.92260    0.65299   1.413 0.157691
## Age                        -0.94444    0.09396 -10.052  < 2e-16 ***
## GenderFemale               -0.70774    0.15712  -4.504 6.66e-06 ***
## Education                  -0.39668    0.07995  -4.961 7.00e-07 ***
## CountryCanada              -0.99488    1.38478  -0.718 0.472486
## CountryNew Zealand         -0.61387    0.32768  -1.873 0.061012 .
## CountryOther                0.42592    0.49694   0.857 0.391398
## CountryIreland              0.35876    0.80181   0.447 0.654557
```

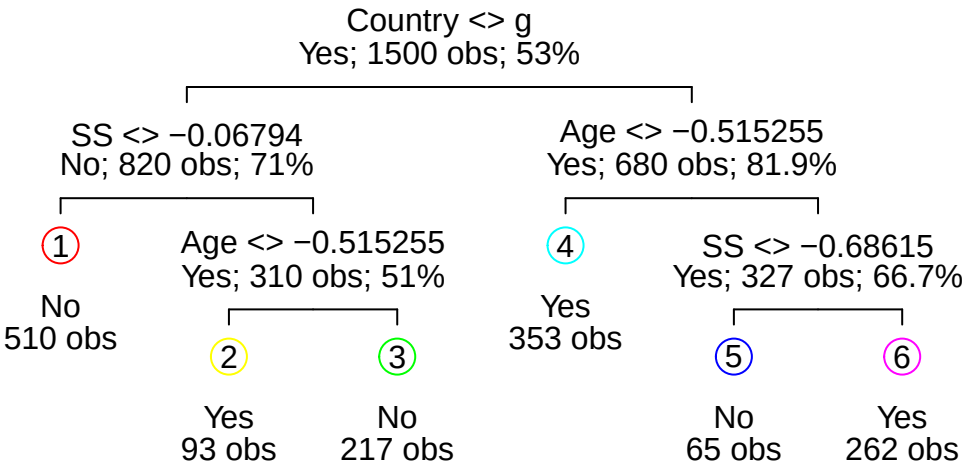
```
## CountryUK                   -0.44685    0.36122  -1.237 0.216066
## CountryUSA                  -1.75153    0.19363  -9.046  < 2e-16 ***
## EthnicityAsian              -1.75981    1.05708  -1.665 0.095956 .
## EthnicityWhite               0.84439    0.64623   1.307 0.191336
## EthnicityMixed:White/Black   0.14902    0.99927   0.149 0.881455
## EthnicityOther               0.73601    0.76088   0.967 0.333388
## EthnicityMixed:White/Asian   0.98784    0.98522   1.003 0.316024
## EthnicityMixed:Black/Asian  12.72270  345.00102   0.037 0.970583
## Nscore                      -0.13356    0.09106  -1.467 0.142438
## Escore                      -0.17346    0.09748  -1.779 0.075164 .
## Oscore                       0.62582    0.09322   6.714 1.90e-11 ***
## Ascore                       0.06894    0.08354   0.825 0.409261
## Cscore                      -0.35025    0.09032  -3.878 0.000105 ***
## Impulsive                   -0.12735    0.10243  -1.243 0.213743
## SS                           0.64533    0.11354   5.684 1.32e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2074  on 1499  degrees of freedom
## Residual deviance: 1173  on 1477  degrees of freedom
## AIC: 1219
##
## Number of Fisher Scoring iterations: 12
```

2.Decision Tree Model

```
#a
set.seed(1)
tree_parameters <- tree.control(nobs=nrow(drug_use_train), minsize=10, mindev=1e-3)
tree_mod <- tree(recent_cannabis_use ~ . , control = tree_parameters, data = drug_use_train)
cvtree <- cv.tree(tree_mod, K = 10, FUN = prune.misclass)
best_size <- min(cvtree$size[which(cvtree$dev == min(cvtree$dev))])
best_tree <- prune.tree(tree_mod, best = best_size)
```

By drawing the tree, we can see that the first split of our tree is by the `Country` variable.

```
#b
draw.tree(best_tree, nodeinfo = TRUE)
```

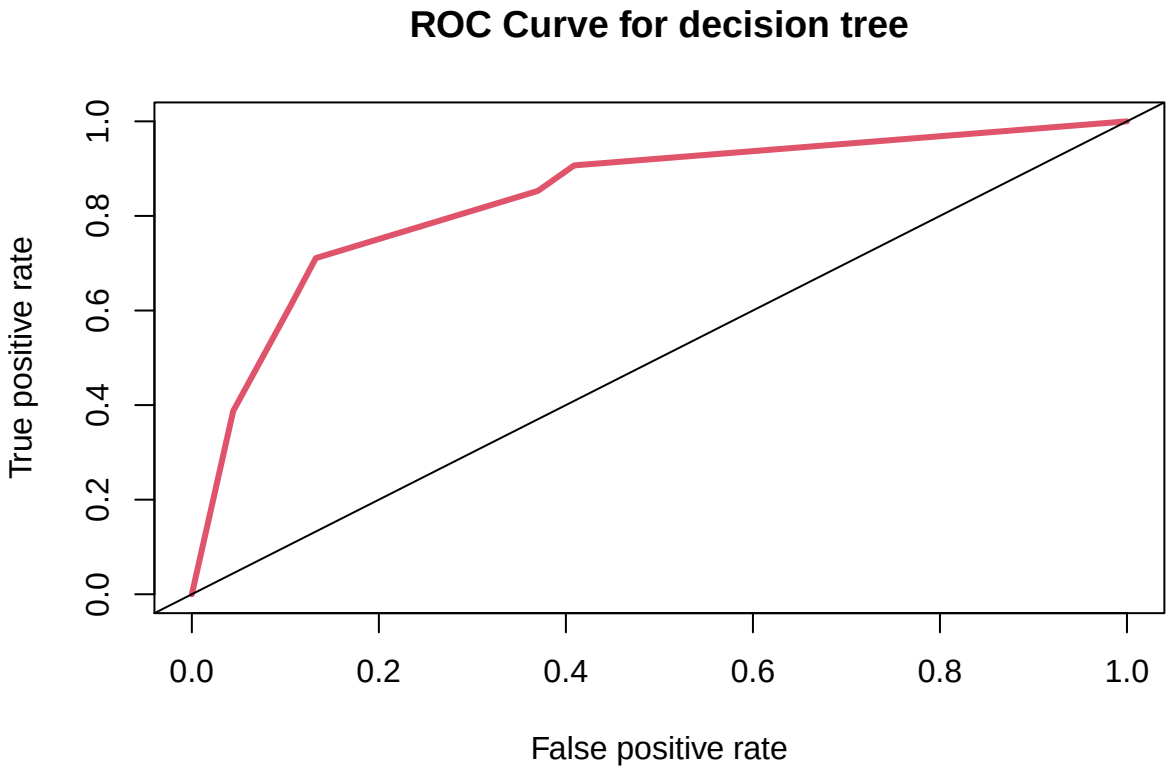


The TPR of our predictions is 0.7867299 and the FPR is 0.1666667. As the *true positive rate* (TPR) is calculated by $\frac{TP}{TP+FN}$, we divide the bottom right element by the second column of our confusion matrix. Likewise, the *false positive rate* (FPR) is calculated by $\frac{FP}{FP+TN}$ which can be obtained by dividing the lower left element by the first column of our confusion matrix.

```
pred.val <- predict(best_tree, drug_use_test, type = 'class')
pred.val1 <- predict(best_tree, drug_use_test, type = 'vector')
err <- table(pred.val, drug_use_test$recent_cannabis_use)
tpr <- err[2,2]/(err[2,2] + err[1,2])
fpr <- err[2,1]/(err[2,1] + err[1,1])
```

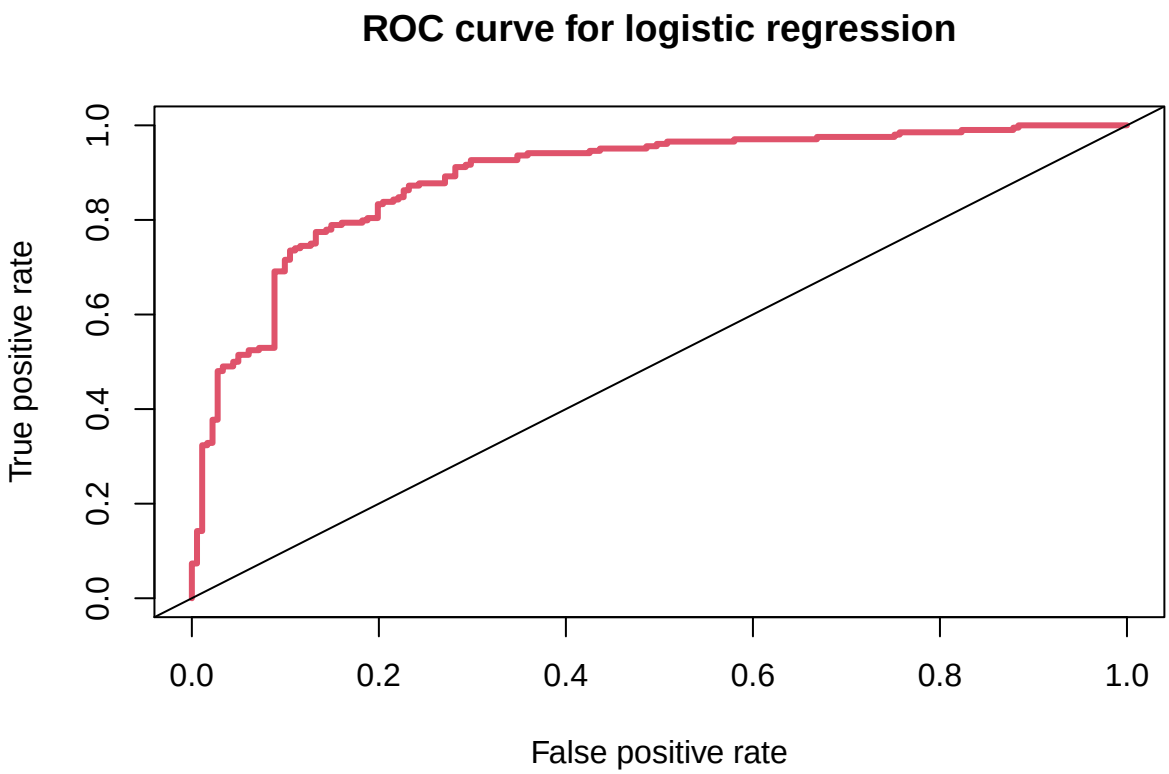
3.ROC curve for decision tree

```
pred <- prediction(pred.val1[,2],drug_use_test$recent_cannabis_use)
perf <- performance(pred, measure = 'tpr',x.measure = 'fpr')
plot(perf, col = 2, lwd = 3, main = 'ROC Curve for decision tree')
abline(0,1)
```



ROC curve for logistic regression

```
pred_log <- predict(log_mod, drug_use_test,type = 'response')
pred2 <- prediction(pred_log, drug_use_test$recent_cannabis_use)
perf2 <- performance(pred2, measure = 'tpr',x.measure = 'fpr')
plot(perf2, col = 2 , lwd = 3 , main = 'ROC curve for logistic regression')
abline(0,1)
```



3(b).Compute AUC for both models and print them. AUC for decision tree is 0.8348046 and AUC for logistic regression is 0.8792247. We can find AUC for logistic regression is larger.

```
auc_tree <- performance(pred,'auc')@y.values
auc_log <- performance(pred2,'auc')@y.values
```

4(a)Convert type column to factor. Use the table command to print the number of patients with each leukemia subtype. Which leukemia subtype occurs the least in this data? BCR-ABL occurs the least in this data.

```
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], 'WAS' => 'WAS_1' [260], 'HLA-
## G' => 'HLA-G_1' [297], 'TRAF3IP3' => 'TRAF3IP3_1' [307], 'ZNF266' =>
## 'ZNF266_1' [364], 'CRYBG1' => 'CRYBG1_1' [441], 'BRD8' => 'BRD8_1' [460], 'MDC1'
## => 'MDC1_1' [464], 'RAC2' => 'RAC2_1' [478], 'IL10RB' => 'IL10RB_1' [483],
## 'AKAP17A' => 'AKAP17A_1' [542], 'N4BP2L1' => 'N4BP2L1_1' [547], 'ARPC4' =>
## 'ARPC4_1' [565], 'SRSF10' => 'SRSF10_1' [576], 'RAPGEF2' => 'RAPGEF2_1' [583],
## 'PARP2' => 'PARP2_1' [587], 'TRIM33' => 'TRIM33_1' [610], 'KAT8' =>
## 'KAT8_1' [665], 'ASMTL' => 'ASMTL_1' [715], 'LSM7' => 'LSM7_1' [727],
## 'HLA-DQB1' => 'HLA-DQB1_1' [732], 'FMR1' => 'FMR1_1' [826], 'RASGRP2' =>
## 'RASGRP2_1' [858], 'LIMK2' => 'LIMK2_1' [866], 'TMEM106C' => 'TMEM106C_1' [881],
## 'TGOLN2' => 'TGOLN2_1' [937], 'SLC25A1' => 'SLC25A1_1' [940], 'NMT1' =>
```

```
## 'NMT1_1' [942], 'ENSA' => 'ENSA_1' [947], 'ENSA' => 'ENSA_2' [948], 'UBR5'
## => 'UBR5_1' [963], 'UBE2J1' => 'UBE2J1_1' [966], 'ACTN1' => 'ACTN1_1' [994],
## 'TRA2A' => 'TRA2A_1' [1003], 'ATXN10' => 'ATXN10_1' [1057], 'CUL1' =>
## 'CUL1_1' [1077], 'XBP1' => 'XBP1_1' [1094], 'ATP2A2' => 'ATP2A2_1' [1110],
## 'LDLRAD4' => 'LDLRAD4_1' [1118], 'ARHGEF2' => 'ARHGEF2_1' [1134],
## 'IDH3B' => 'IDH3B_1' [1141], 'SERBP1' => 'SERBP1_1' [1188], 'TRIM44' =>
## 'TRIM44_1' [1205], 'TRIM44' => 'TRIM44_2' [1206], 'PTPRC' => 'PTPRC_1' [1219],
## 'PTPRC' => 'PTPRC_2' [1220], 'PPP2R5C' => 'PPP2R5C_1' [1235], 'PPP2R5C'
## => 'PPP2R5C_2' [1236], 'ADAM10' => 'ADAM10_1' [1241], 'NFATC3' =>
## 'NFATC3_1' [1252], 'ILF3' => 'ILF3_1' [1264], 'RBM6' => 'RBM6_1' [1274],
## 'CTNNA1' => 'CTNNA1_1' [1297], 'CTNNA1' => 'CTNNA1_2' [1298], 'IGHM' =>
## 'IGHM_1' [1302], 'IGHM' => 'IGHM_2' [1303], 'IGHM' => 'IGHM_3' [1304], 'SFPQ' =>
## 'SFPQ_2' [1321], 'RBCK1' => 'RBCK1_1' [1398], 'NFATC2IP' => 'NFATC2IP_1' [1408],
## 'ILF3' => 'ILF3_2' [1432], 'RAE1' => 'RAE1_1' [1436], 'ITPR1' =>
## 'ITPR1_1' [1443], 'NCBP2' => 'NCBP2_1' [1448], 'STAT1' => 'STAT1_1' [1486],
## 'AZIN1' => 'AZIN1_1' [1497], 'SEC13' => 'SEC13_1' [1517], 'ABI1' =>
## 'ABI1_1' [1565], 'CYB5B' => 'CYB5B_1' [1607], 'HUWE1' => 'HUWE1_1' [1624],
## 'RAB1A' => 'RAB1A_1' [1634], 'AHCYL1' => 'AHCYL1_1' [1652], 'EIF1AX' =>
## 'EIF1AX_1' [1661], 'MAGED2' => 'MAGED2_1' [1689], 'SCAF11' => 'SCAF11_1' [1709],
## 'BLCAP' => 'BLCAP_1' [1716], 'TROVE2' => 'TROVE2_1' [1729], 'CTCF' =>
## 'CTCF_1' [1745], 'RAB8A' => 'RAB8A_1' [1754], 'ACTR2' => 'ACTR2_1' [1768],
## 'HMGN4' => 'HMGN4_1' [1771], 'NDUFB7' => 'NDUFB7_1' [1793], 'VAMP3' =>
## 'VAMP3_1' [1796], 'SRSF6' => 'SRSF6_1' [1808], 'TNPO3' => 'TNPO3_1' [1811],
## 'SRSF1' => 'SRSF1_1' [1834], 'TMED10' => 'TMED10_1' [1847], 'AP3D1' =>
## 'AP3D1_1' [1872], 'MAPKAPK2' => 'MAPKAPK2_1' [1877], 'BRD2' => 'BRD2_1' [1891],
## 'BRD2' => 'BRD2_2' [1892], 'GARS' => 'GARS_1' [1901], 'SNX1' => 'SNX1_1' [1902],
## 'TSC22D3' => 'TSC22D3_1' [1927], 'AMD1' => 'AMD1_1' [1951], 'LITAF' =>
## 'LITAF_1' [2011], 'GLUD1' => 'GLUD1_1' [2059], 'KDELR1' => 'KDELR1_1' [2079],
## 'PGK1' => 'PGK1_1' [2099], 'VDAC2' => 'VDAC2_1' [2107], 'ADH5' =>
## 'ADH5_1' [2111], 'MEF2C' => 'MEF2C_1' [2113], 'MEF2C' => 'MEF2C_2' [2114],
## 'RCN2' => 'RCN2_1' [2125], 'PCMT1' => 'PCMT1_1' [2134], 'PCMT1' =>
## 'PCMT1_2' [2135], 'CD79A' => 'CD79A_1' [2149], 'MARCH6' => 'MARCH6_1' [2169],
## 'CBX3' => 'CBX3_1' [2180], 'LSM14A' => 'LSM14A_1' [2217], 'SORL1' =>
## 'SORL1_1' [2220], 'ICAM2' => 'ICAM2_1' [2244], 'SNRPB' => 'SNRPB_1' [2246],
## 'CYB5A' => 'CYB5A_1' [2248], 'BTN3A2' => 'BTN3A2_1' [2277], 'DICER1' =>
## 'DICER1_1' [2280], 'HADH' => 'HADH_1' [2281], 'HDGF' => 'HDGF_1' [2285], 'SEPT6'
## => 'SEPT6_2' [2306], 'SSBP1' => 'SSBP1_1' [2315], 'H2AFV' => 'H2AFV_1' [2318],
## 'PTPA' => 'PTPA_1' [2331], 'FBL' => 'FBL_1' [2354], 'OGT' => 'OGT_1' [2362],
## 'SLC25A1' => 'SLC25A1_2' [2377], 'FUBP1' => 'FUBP1_1' [2386], 'TUBGCP2' =>
## 'TUBGCP2_1' [2400], 'COX5B' => 'COX5B_1' [2402], 'VDAC1' => 'VDAC1_1' [2410],
## 'HNRNPDL' => 'HNRNPDL_1' [2431], 'THUMPD1' => 'THUMPD1_1' [2443], 'CDV3'
## => 'CDV3_1' [2444], 'UBE3B' => 'UBE3B_1' [2447], 'SFPQ' => 'SFPQ_3' [2451],
## 'STX16' => 'STX16_1' [2452], 'SMARCA2' => 'SMARCA2_1' [2471], 'CHD8' =>
## 'CHD8_1' [2475], 'TCF25' => 'TCF25_1' [2490], 'API5' => 'API5_1' [2491],
## 'SAP18' => 'SAP18_1' [2493], 'AHCYL1' => 'AHCYL1_2' [2501], 'CTBP1' =>
## 'CTBP1_1' [2503], 'AES' => 'AES_1' [2512], 'PURA' => 'PURA_1' [2514], 'BCL11A'
## => 'BCL11A_1' [2518], 'BUB3' => 'BUB3_1' [2534], 'RER1' => 'RER1_1' [2537],
## 'ATXN2L' => 'ATXN2L_1' [2541], 'JAK1' => 'JAK1_1' [2548], 'GUSBP11' =>
## 'GUSBP11_1' [2564], 'JTB' => 'JTB_1' [2568], 'BRD3' => 'BRD3_1' [2571], 'RSU1'
## => 'RSU1_1' [2584], 'ADD3' => 'ADD3_1' [2619], 'UBE2I' => 'UBE2I_1' [2627],
## 'MRPS12' => 'MRPS12_1' [2640], 'CTNNA1' => 'CTNNA1_3' [2641], 'XRCC5' =>
## 'XRCC5_1' [2642], 'ITGA4' => 'ITGA4_1' [2644], 'CTNNA1' => 'CTNNA1_4' [2647],
## 'FYN' => 'FYN_1' [2649], 'ERG' => 'ERG_1' [2652], 'RAC1' => 'RAC1_1' [2654],
## 'LCK' => 'LCK_1' [2657], 'PTK2B' => 'PTK2B_1' [2664], 'SKP1' =>
```

```
## 'SKP1_1' [2665], 'PRKDC' => 'PRKDC_1' [2666], 'MYC' => 'MYC_1' [2668], 'RBL2'
## => 'RBL2_1' [2673], 'AZIN1' => 'AZIN1_2' [2674], 'CCNA2' => 'CCNA2_1' [2681],
## 'FOS' => 'FOS_1' [2688], 'FOS' => 'FOS_2' [2689], 'RAF1' => 'RAF1_1' [2690],
## 'RAP1B' => 'RAP1B_1' [2692], 'ERCC1' => 'ERCC1_1' [2696], 'ERCC1' =>
## 'ERCC1_2' [2697], 'RAN' => 'RAN_1' [2702], 'TRIM27' => 'TRIM27_1' [2703],
## 'PMS2P3' => 'PMS2P3_1' [2708], 'TGFBR2' => 'TGFBR2_1' [2710], 'PCNA' =>
## 'PCNA_1' [2712], 'MYC' => 'MYC_2' [2714], 'CDK13' => 'CDK13_1' [2717],
## 'CCND3' => 'CCND3_1' [2719], 'FARSA' => 'FARSA_1' [2732], 'FARSA' =>
## 'FARSA_2' [2733], 'DAXX' => 'DAXX_1' [2734], 'UBE3A' => 'UBE3A_1' [2735],
## 'ARAF' => 'ARAF_1' [2739], 'UBE2N' => 'UBE2N_1' [2747], 'RASA1' =>
## 'RASA1_1' [2748], 'ABL1' => 'ABL1_1' [2749], 'ABL1' => 'ABL1_2' [2750], 'MTA1'
## => 'MTA1_1' [2753], 'EIF3I' => 'EIF3I_1' [2754], 'SYK' => 'SYK_1' [2761],
## 'TOP2A' => 'TOP2A_1' [2762], 'RB1' => 'RB1_1' [2764], 'TOP2B' =>
## 'TOP2B_1' [2765], 'TNFRSF1B' => 'TNFRSF1B_1' [2766], 'GRB2' => 'GRB2_1' [2769],
## 'RBM5' => 'RBM5_1' [2770], 'N4BP2L1' => 'N4BP2L1_2' [2773], 'N4BP2L2' =>
## 'N4BP2L2_1' [2774], 'NME1' => 'NME1_1' [2775], 'TYMS' => 'TYMS_1' [2776],
## 'DYRK1A' => 'DYRK1A_1' [2778], 'FEN1' => 'FEN1_1' [2779], 'FEN1' =>
## 'FEN1_2' [2780], 'ETS2' => 'ETS2_1' [2781], 'FNTA' => 'FNTA_1' [2783], 'JAK1'
## => 'JAK1_2' [2787], 'MYB' => 'MYB_1' [2792], 'MYB' => 'MYB_2' [2793], 'MYB' =>
## 'MYB_3' [2794], 'MYB' => 'MYB_4' [2795], 'MYB' => 'MYB_5' [2796], 'SMAD2' =>
## 'SMAD2_1' [2798], 'PTEN' => 'PTEN_1' [2799], 'MAPKAPK2' => 'MAPKAPK2_2' [2800],
## 'PSMD9' => 'PSMD9_1' [2801], 'PSMA4' => 'PSMA4_1' [2806], 'SRF' =>
## 'SRF_1' [2810], 'LYN' => 'LYN_1' [2815], 'IL7R' => 'IL7R_1' [2817], 'TCF3' =>
## 'TCF3_1' [2818], 'TCF3' => 'TCF3_2' [2819], 'NFKB1' => 'NFKB1_1' [2820], 'NFKB1'
## => 'NFKB1_2' [2821], 'RPA1' => 'RPA1_1' [2822], 'PPP2R2A' => 'PPP2R2A_1' [2823],
## 'TERF1' => 'TERF1_1' [2826], 'BCR' => 'BCR_1' [2828], 'RBBP4' =>
## 'RBBP4_1' [2830], 'TERF2' => 'TERF2_1' [2831], 'PSMB4' => 'PSMB4_1' [2834],
## 'PSMB7' => 'PSMB7_1' [2836], 'PARP1' => 'PARP1_1' [2838], 'RELA' =>
## 'RELA_1' [2840], 'RELA' => 'RELA_2' [2841], 'EIF2S3' => 'EIF2S3_1' [2842],
## 'YWHAZ' => 'YWHAZ_1' [2846], 'PTP4A2' => 'PTP4A2_1' [2847], 'POLR2H' =>
## 'POLR2H_1' [2850], 'GAB1' => 'GAB1_2' [2851], 'PRKDC' => 'PRKDC_2' [2852],
## 'PRKCB' => 'PRKCB_1' [2855], 'SAT1' => 'SAT1_1' [2862], 'PTPRE' =>
## 'PTPRE_1' [2865], 'RPL22' => 'RPL22_1' [2866], 'EIF2S1' => 'EIF2S1_1' [2867],
## 'CYC1' => 'CYC1_1' [2869], 'HSP90AB1' => 'HSP90AB1_2' [2870], 'CD44' =>
## 'CD44_1' [2873], 'MAP2K1' => 'MAP2K1_1' [2875], 'TNK2' => 'TNK2_1' [2877],
## 'GNA13' => 'GNA13_1' [2879], 'NR3C1' => 'NR3C1_1' [2882], 'RAB1A' =>
## 'RAB1A_2' [2888], 'ODC1' => 'ODC1_1' [2890], 'PLCG2' => 'PLCG2_1' [2891], 'RFC4'
## => 'RFC4_1' [2894], 'FLT3' => 'FLT3_1' [2895], 'EIF2AK2' => 'EIF2AK2_1' [2902],
## 'USP9X' => 'USP9X_1' [2913], 'PSMD7' => 'PSMD7_1' [2917], 'PPP1CA' =>
## 'PPP1CA_1' [2924], 'TUBB4B' => 'TUBB4B_2' [2926], 'ARRB2' => 'ARRB
```

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

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

```
## # A tibble: 327 x 3,142
```

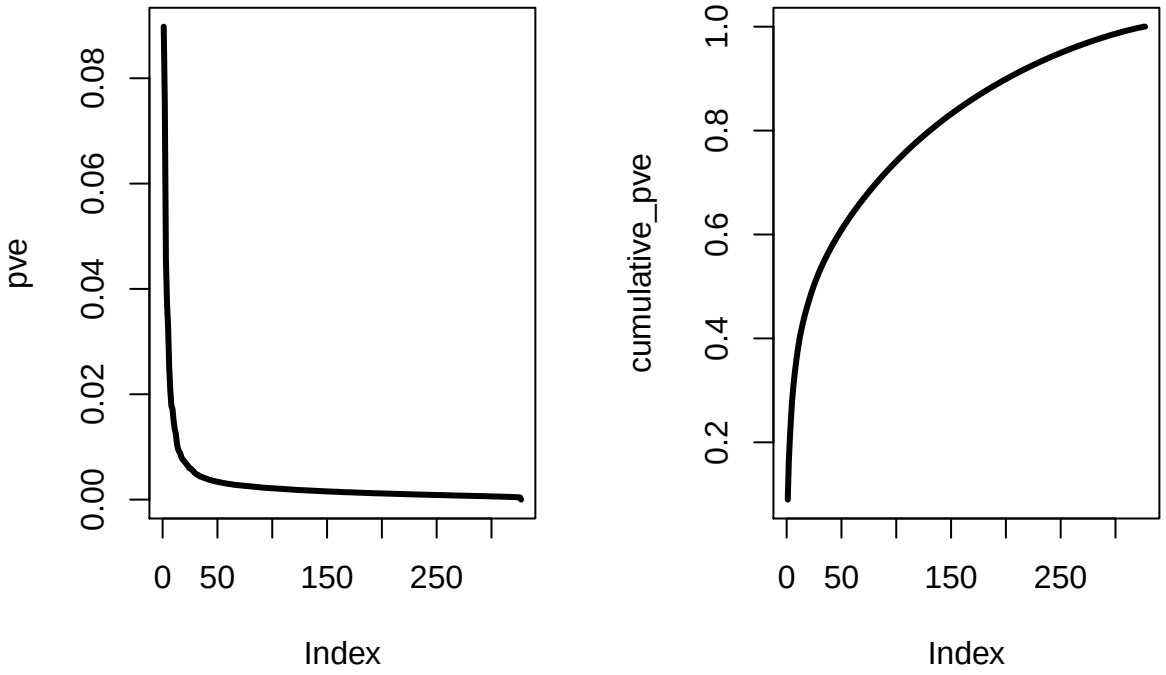
```
##    Type    FCGRT FCGRT_1 `31444_s_at` TMSB10  PGK1 EIF3K `31503_at` HDLBP TXNIP
##    <fct>   <dbl>   <dbl>        <dbl>  <dbl> <dbl> <dbl>      <dbl> <dbl> <dbl>
##  1 BCR-ABL  8.54    9.43         10.9    10.5  8.07  9.62       8.86  8.59 10.6
##  2 BCR-ABL  8.20    9.59         10.3    10.7  7.86  9.88       8.48  8.55 10.6
##  3 BCR-ABL  8.38    8.94         10.6    10.5  7.39  9.69       8.16  8.51  9.88
##  4 BCR-ABL  8.07    9.60         10.4    10.8  7.73 10.1        8.82  8.76 10.4
##  5 BCR-ABL  8.65    8.42          9.85   10.3  8.22  9.77       8.68  8.32 10.3
##  6 BCR-ABL 10.1    10.6          10.5    10.4  7.83  9.81       8.82  9.10  9.71
##  7 BCR-ABL  8.36    9.71         10.3    10.6  7.19 10.3        8.62  8.35 10.3
##  8 BCR-ABL  8.21    8.61          9.92   10.6  8.77  9.88       8.43  8.82 10.2
##  9 BCR-ABL  8.92    9.40         10.1    10.7  8.11  9.31       8.13  8.28 10.2
## 10 BCR-ABL  8.09    9.07         10.7    10.8  7.38  9.86       8.07  8.49 11.1
## # ... with 317 more rows, and 3,132 more variables: 31510_s_at <dbl>,
## #   31519_f_at <dbl>, 31522_f_at <dbl>, HIST1H2BE <dbl>, 31524_f_at <dbl>,
## #   31526_f_at <dbl>, HIST1H2BM <dbl>, RTN4 <dbl>, GUSBP11 <dbl>,
## #   31600_s_at <dbl>, VDAC1 <dbl>, NDUFS7 <dbl>, SRP72 <dbl>, 31673_s_at <dbl>,
## #   TOP1P2 <dbl>, GLUD1 <dbl>, GPR35 <dbl>, HSBP1 <dbl>, BTF3 <dbl>,
## #   DDX11 <dbl>, MARF1 <dbl>, MT4 <dbl>, 31993_f_at <dbl>, 32004_s_at <dbl>,
## #   32007_at <dbl>, 32408_s_at <dbl>, RPL15 <dbl>, ATP6V0E2 <dbl>, SUMO4 <dbl>,
## #   TSPO2 <dbl>, OR2B6 <dbl>, S1PR4 <dbl>, TXNL4A <dbl>, GNA13 <dbl>,
## #   AIF1 <dbl>, GM2A <dbl>, HNRNPC <dbl>, TUBB4B <dbl>, TUBB4B_1 <dbl>,
## #   TUG1 <dbl>, 33689_s_at <dbl>, LONRF1 <dbl>, IGSF9B <dbl>, VIM <dbl>,
## #   34093_at <dbl>, 34099_f_at <dbl>, IGHM <dbl>, HIST1H2AL <dbl>,
## #   SLC6A7 <dbl>, DNTT <dbl>, AKAP17A <dbl>, EFNA3 <dbl>, FLT3 <dbl>,
## #   YWHAZ <dbl>, 34647_at <dbl>, SSR1 <dbl>, SSR1_1 <dbl>, COMT <dbl>,
## #   HLA-J <dbl>, ZNF254 <dbl>, ZNF273 <dbl>, DCUN1D4 <dbl>, GPRIN2 <dbl>,
## #   35566_f_at <dbl>, ZNF253 <dbl>, HIST1H2BL <dbl>, LY9 <dbl>,
## #   HIST1H2BN <dbl>, ERG <dbl>, CTCF <dbl>, IRF7 <dbl>, HDGF <dbl>,
## #   PMS2P1 <dbl>, HSP90AA1 <dbl>, 32317_s_at <dbl>, HLA-E <dbl>, YWHAB <dbl>,
## #   IDH2 <dbl>, ALDOA <dbl>, PNMT <dbl>, PKM <dbl>, MRE11 <dbl>,
## #   32872_at <dbl>, TCF3 <dbl>, 32877_i_at <dbl>, 32878_f_at <dbl>,
## #   PTPRE <dbl>, 32921_at <dbl>, RIF1 <dbl>, FCMR <dbl>, RIPOR2 <dbl>,
## #   GAB1 <dbl>, 32980_f_at <dbl>, 33458_r_at <dbl>, 33499_s_at <dbl>,
## #   33500_i_at <dbl>, 33501_r_at <dbl>, HBD <dbl>, P2RX1 <dbl>, PNN <dbl>, ...
```

```
table(leukemia_data$Type)
```

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

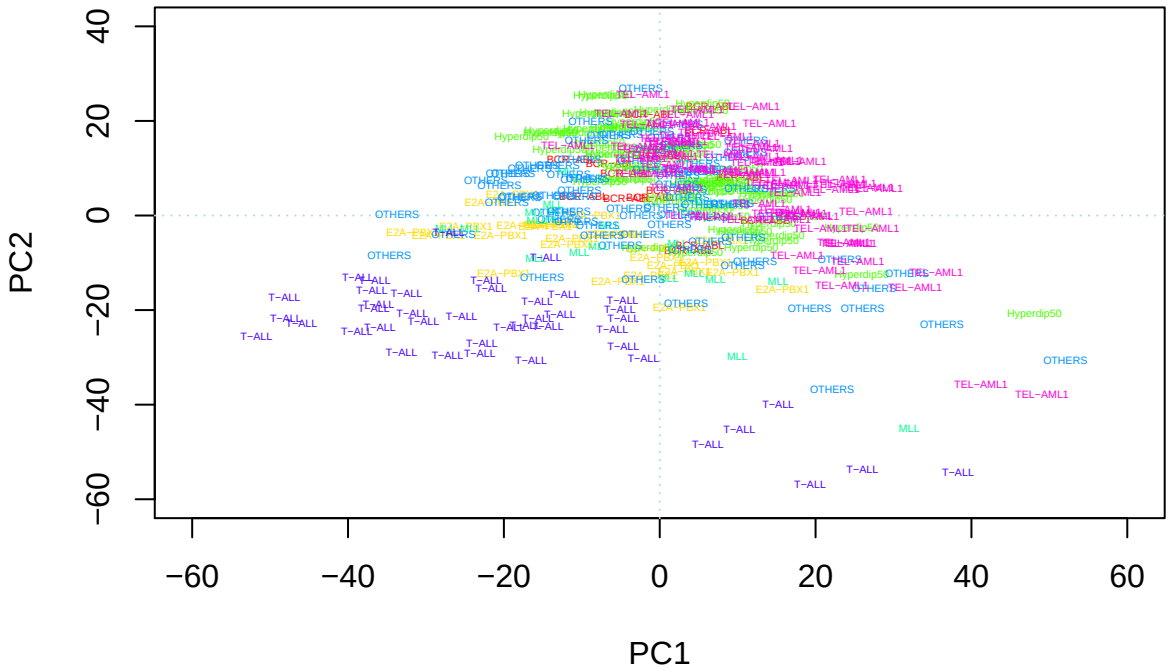
Run PCA on the leukemia data using prcomp function with scale=TRUE and center=TRUE (this scales each gene to have mean 0 and variance 1).

```
pr.out <- prcomp(leukemia_data %>% select(-Type), scale = TRUE, center = TRUE)
pr.var <- pr.out$sdev ^ 2
pve <- pr.var / sum(pr.var)
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)
```



4c.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$Type]
new_coords <- pr.out$x[,1:2]
plot(new_coords, xlim = c(-60,60), ylim = c(-60,40), cex = 0 , main = 'PC1 & PC2')
text(-new_coords, label = leukemia_data$Type, cex = 0.3, col = plot_colors)
abline(h = 0, v = 0, col = 'lightblue', lty = 3)
```



T-ALL is most clearly separated from the others along the PC1 axis.

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

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

4f.

```
library(dendextend)
```

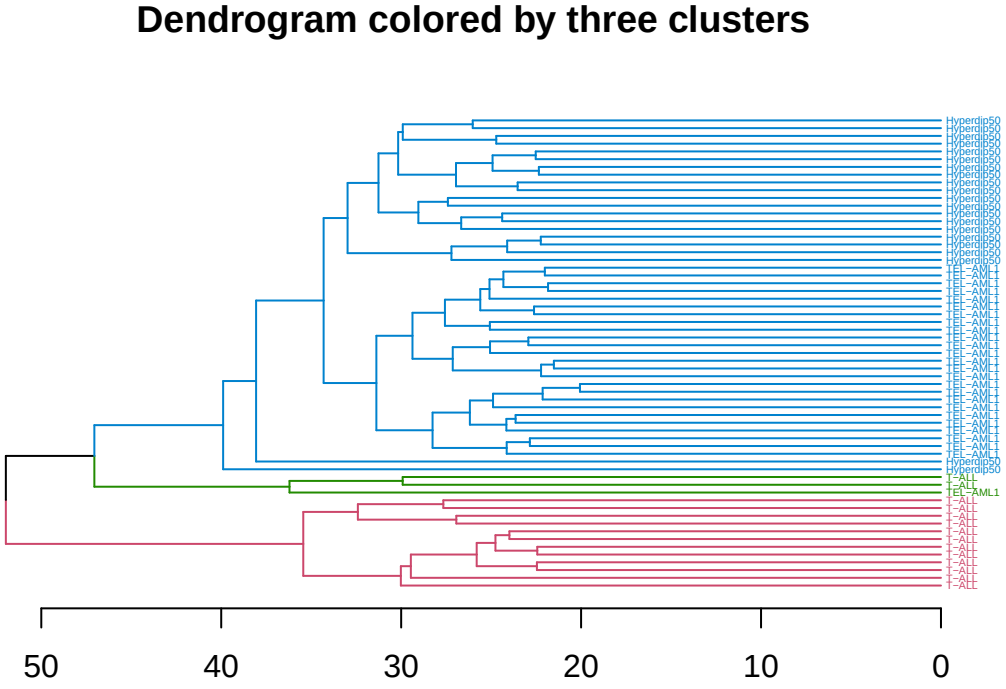
```
##
## ---------------------
## Welcome to dendextend version 1.15.1
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports 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
```

```
leukemia_subset <- leukemia_data %>% filter(Type == c('T-ALL', 'TEL-AML1', 'Hyperdip50'))
dis <- dist(leukemia_subset[,-1], method = 'euclidean')
leukemia.hc <- hclust(dis, method = 'complete')
dend1 <- as.dendrogram(leukemia.hc)
dend1 <- color_branches(dend1, k = 3)
dend1 <- color_labels(dend1, k = 3)
dend1 <- set(dend1, 'labels_cex', 0.3)
dend1 <- set_labels(dend1, labels = leukemia_subset$Type[order.dendrogram(dend1)])
plot(dend1, horiz = T, main = 'Dendrogram colored by three clusters')
```




```
dend2 <- as.dendrogram(leukemia.hc)
dend2 <- color_branches(dend2, k = 5)
dend2 <- color_labels(dend2, k = 5)
dend2 <- set(dend2, 'labels_cex', 0.3)
dend2 <- set_labels(dend2, labels = leukemia_subset$Type[order.dendrogram(dend2)])
plot(dend2, horiz = T, main = 'Dendrogram colored by five clusters')
```

Dendrogram colored by five clusters