Homework 3

Justin Lau

11/12/2020

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──
```

```
## ✓ ggplot2 3.3.2     ✓ purrr   0.3.4
## ✓ tibble  3.0.4     ✓ dplyr   1.0.2
## ✓ tidyr   1.1.2     ✓ stringr 1.4.0
## ✓ readr   1.4.0     ✓ forcats 0.5.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ROCR)
library(tree)
```

```
## Registered S3 method overwritten by 'tree':
##   method     from
##   print.tree cli
```

```
library(maptree)
```

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

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

```
library(class)
library(lattice)
library(ggridges)
library(superheat)
library(ggplot2)
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/dend
extend/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
```

```
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','Ecst
asy','Heroin','Ketamine',
'Legalh','LSD','Meth','Mushrooms','Nicotine','Semer','VSA'))
```

```
##
## ── 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")))
```

Question 1a

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

1b

```
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
```

1c

```
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
## -2.6109  -0.5263  -0.1403   0.5883   3.0188
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -1.08560    0.65907  -1.647  0.09953 .
## Age                          0.86520    0.09475   9.132  < 2e-16 ***
## GenderFemale                 0.73614    0.15957   4.613 3.96e-06 ***
## Education                    0.36791    0.08074   4.557 5.19e-06 ***
## CountryCanada              -12.90061  627.10873  -0.021  0.98359
## CountryNew Zealand           1.03173    0.33373   3.091  0.00199 **
## CountryOther                -0.11479    0.49598  -0.231  0.81697
## CountryIreland               0.61605    0.75728   0.814  0.41593
## CountryUK                    0.71993    0.37124   1.939  0.05247 .
## CountryUSA                   2.00336    0.19797  10.119  < 2e-16 ***
## EthnicityAsian               1.01655    0.96363   1.055  0.29146
## EthnicityWhite              -0.91667    0.65040  -1.409  0.15872
## EthnicityMixed:White/Black  -0.38753    1.07561  -0.360  0.71863
## EthnicityOther              -1.07930    0.78116  -1.382  0.16707
## EthnicityMixed:White/Asian  -0.97606    0.99208  -0.984  0.32519
## EthnicityMixed:Black/Asian -14.28533  760.35463  -0.019  0.98501
## Nscore                       0.07098    0.09237   0.768  0.44225
## Escore                       0.07424    0.09666   0.768  0.44244
## Oscore                      -0.71596    0.09488  -7.546 4.49e-14 ***
## Ascore                      -0.08331    0.08145  -1.023  0.30635
## Cscore                       0.34612    0.09168   3.775  0.00016 ***
## Impulsive                    0.07823    0.10539   0.742  0.45789
## SS                          -0.51292    0.11515  -4.454 8.41e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2074.3  on 1499  degrees of freedom
## Residual deviance: 1170.5  on 1477  degrees of freedom
## AIC: 1216.5
##
## Number of Fisher Scoring iterations: 14
```

Question 2

```
set.seed(1)
tree_parameters = tree.control(nobs=nrow(drug_use_train), minsize=10, mindev=1e-3)
drugtree <- tree(recent_cannabis_use~., drug_use_train, control=tree_parameters)
cv <- cv.tree(drugtree, FUN =prune.misclass, K=10 )
cv
```

```
## $size
##  [1] 124  86  78  70  67  62  54  51  48  44  38  33  28  21  16  11   7   5   4
## [20]   2   1
##
## $dev
##  [1] 304 295 295 295 295 295 295 295 295 295 295 295 295 295 295 295 295 308 310
## [20] 350 706
##
## $k
##  [1]        -Inf   0.0000000   0.2500000   0.5000000   0.6666667   0.8000000
##  [7]   1.0000000   1.3333333   1.6666667   1.7500000   2.0000000   2.2000000
## [13]   2.4000000   2.4285714   3.0000000   3.2000000   3.2500000   8.0000000
## [19]  14.0000000  24.0000000 356.0000000
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
best_size = min(cv$size[cv$dev ==315])
```

```
## Warning in min(cv$size[cv$dev == 315]): no non-missing arguments to min;
## returning Inf
```

```
best_size
```

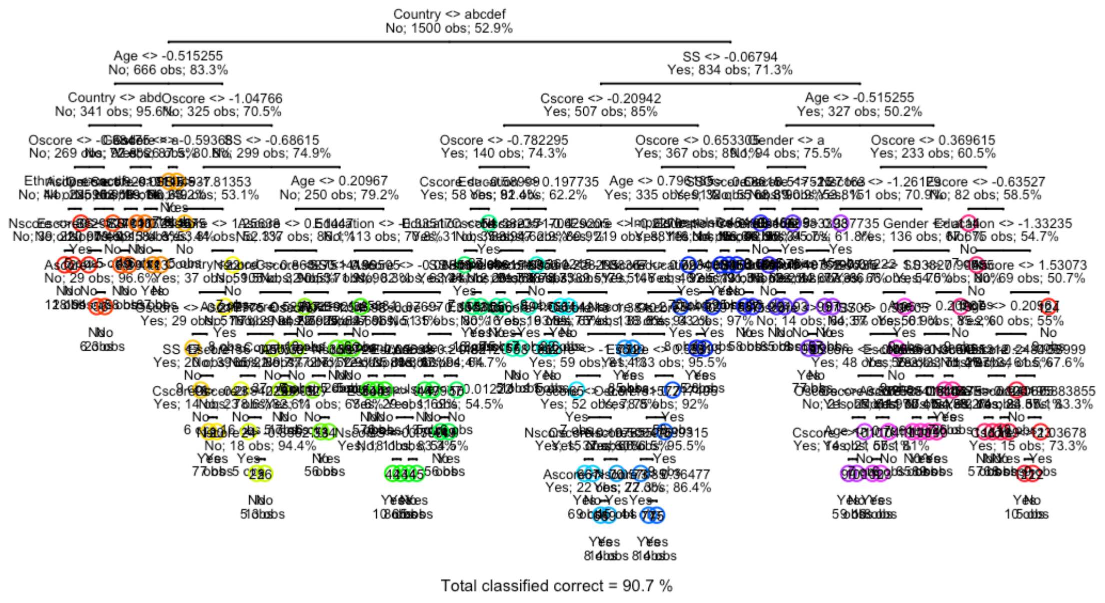
```
## [1] Inf
```

2b

```
drugtree.pruned <- prune.misclass(drugtree, best = best_size)
```

```
## Warning in prune.tree(tree = drugtree, best = best_size, method = "misclass"):
## best is bigger than tree size
```

```
draw.tree(drugtree.pruned, nodeinfo = TRUE, cex = 0.4)
```



2c

```
drugtree_test <- tree(recent_cannabis_use~., drug_use_test, control=tree_parameters)
prob.training = predict(drugtree_test, drug_use_test, type="class")
table(true=drug_use_test$recent_cannabis_use, pred=prob.training)
```

```
##       pred
## true   No Yes
##   No  175  30
##   Yes  11 169
```

```
TPR <- 193/(193+20)
FPR <- 19/(19+153)
cat('TPR is', TPR, '\n')
```

```
## TPR is 0.9061033
```

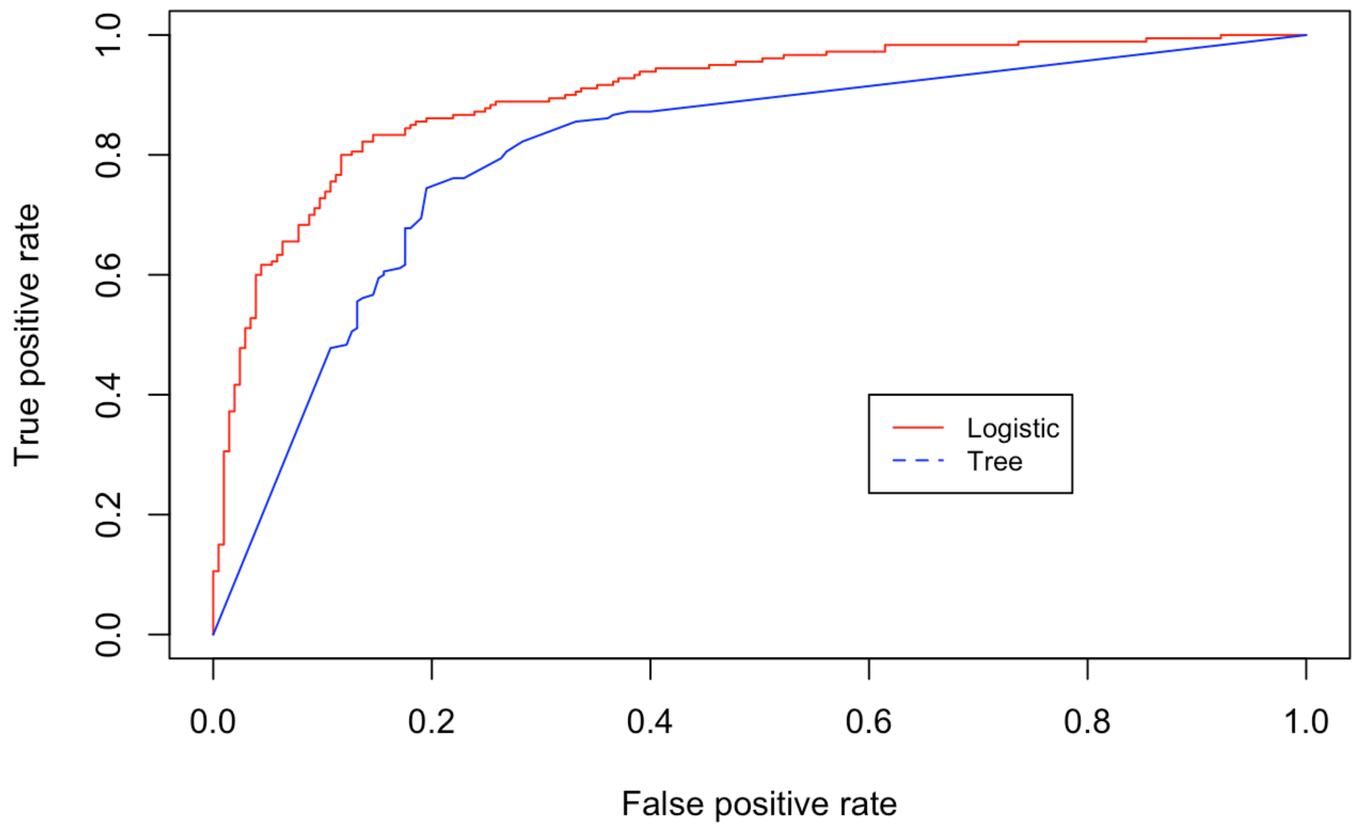
```
cat('FPR is', FPR)
```

```
## FPR is 0.1104651
```

Question 3

```
#Logisitic Regression
fit.test <- glm(recent_cannabis_use~., data=drug_use_test, family=binomial)
training = predict(fit.test, type="response")
pred = prediction(training, drug_use_test$recent_cannabis_use)
perf_log = performance(pred, measure="tpr", x.measure="fpr")
plot(perf_log, col='red')

pred_tree <- predict(drugtree.pruned,drug_use_test, type="vector")
tree.pred <- prediction(pred_tree[,2], drug_use_test$recent_cannabis_use)
perf_tree = performance(tree.pred, measure="tpr", x.measure="fpr")
plot(perf_tree, col='blue',  add=TRUE)
legend(.6,.4, legend=c("Logistic", "Tree"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



3b

```
auc_log = performance(pred, "auc")@y.values
auc_tree = performance(tree.pred, "auc")@y.values

cat("The AUC of logistic is", as.numeric(auc_log), '\n')
```

```
## The AUC of logistic is 0.9020867
```

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

```
## The AUC of decision tree is 0.8067615
```

Question 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], '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 = factor(Type))

leukemia_data %>% count(Type, sort = TRUE)
```

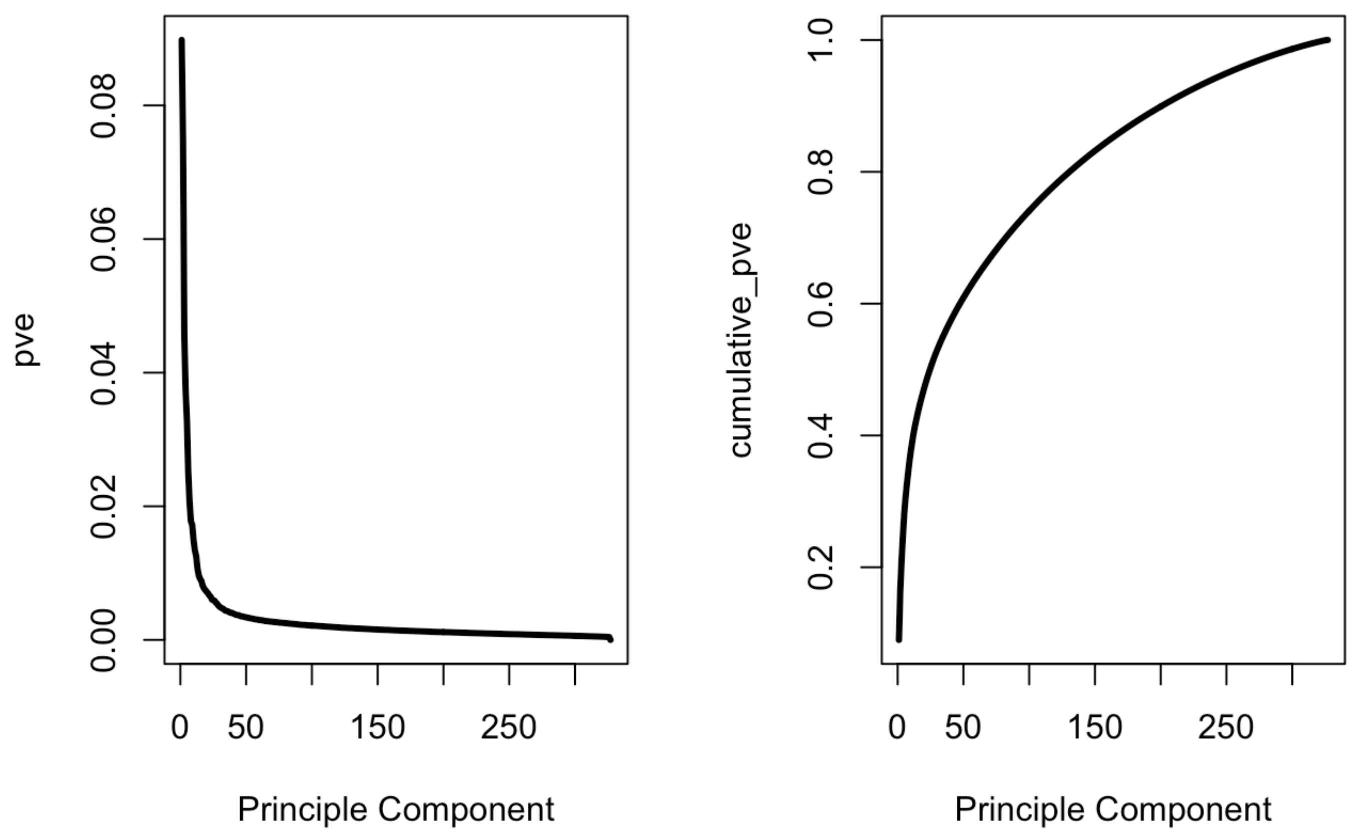
```
## # A tibble: 7 x 2
##   Type            n
##   <fct>       <int>
## 1 OTHERS         79
## 2 TEL-AML1       79
## 3 Hyperdip50     64
## 4 T-ALL          43
## 5 E2A-PBX1       27
## 6 MLL            20
## 7 BCR-ABL        15
```

BCR-ABL occurs the least in the data

4b

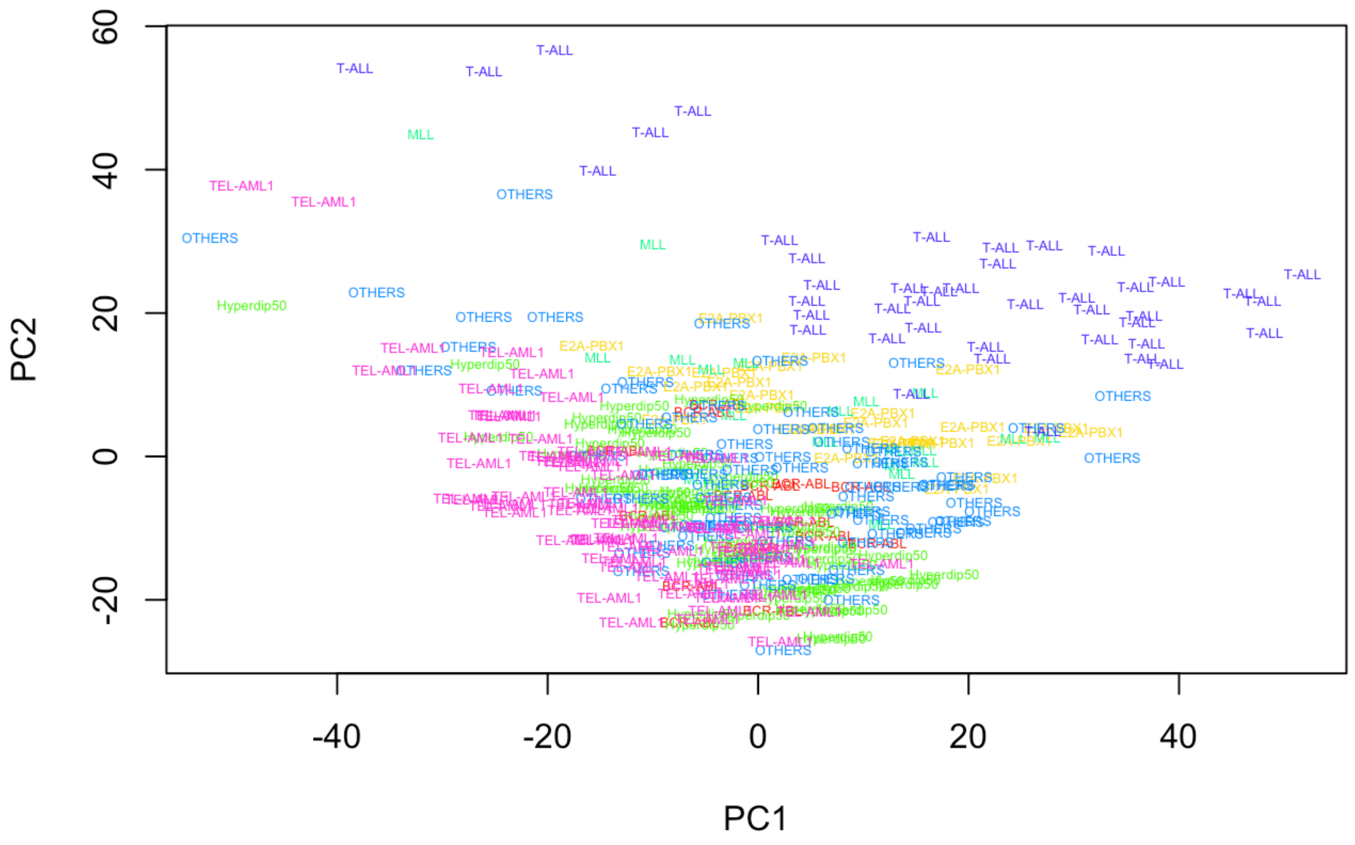
```
leukemia <- leukemia_data %>% select(-Type)

pca <- prcomp(leukemia,scale=TRUE, center=TRUE)
sdev <- pca$sdev
pve <- sdev^2 / sum(sdev^2)
cumulative_pve <- cumsum(pve)
par(mfrow=c(1, 2))
plot(pve, type="l", xlab = "Principle Component",lwd=3)
plot(cumulative_pve, type="l", xlab = "Principle Component", lwd=3)
```



4c

```
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, cex=0.4, col=plot_colors)
```



T-

ALL seems to be separated the most with regards to PC1

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

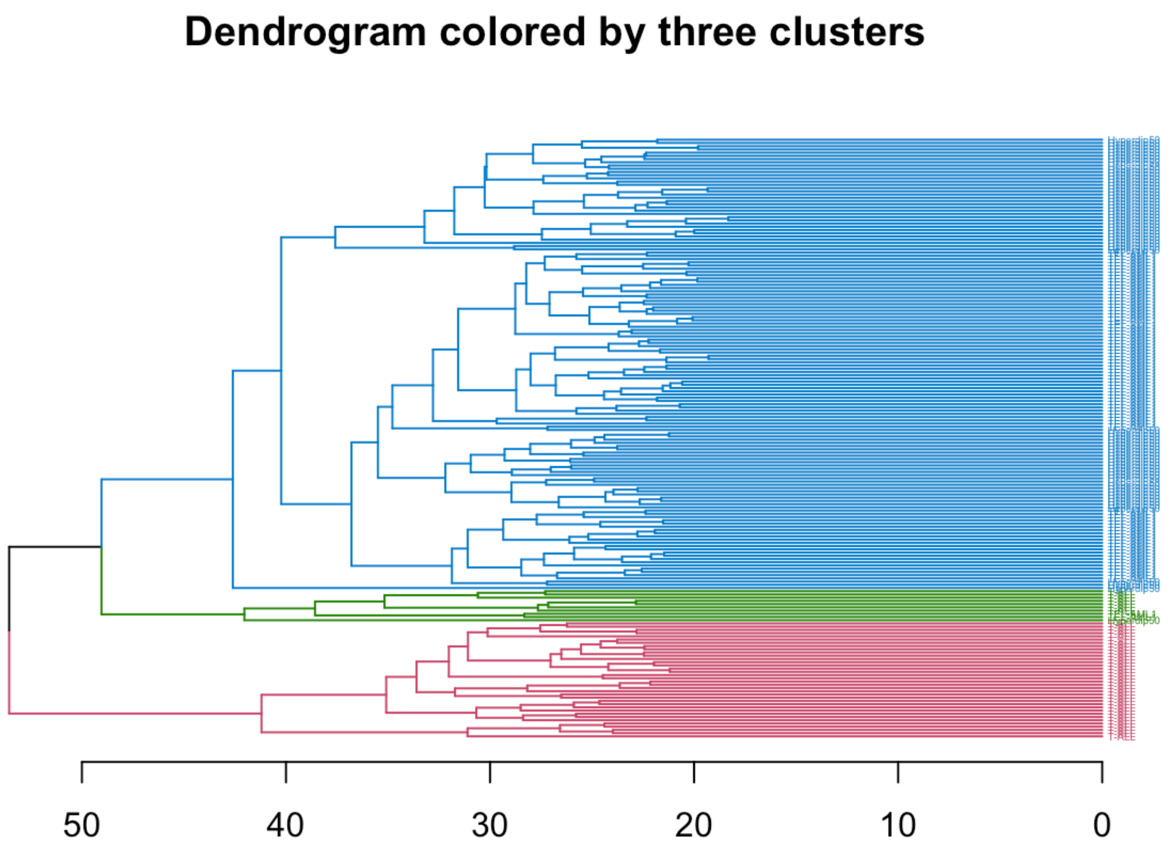
```
##        SRSF8        BUB1B       SEC11A     35985_at        EVI2B       ZFAND5
## 7.950999e-07 3.499181e-06 2.400636e-05 3.193166e-05 3.282533e-05 3.513191e-05
```

4f

```
leukemia_subset <- leukemia_data %>% filter(Type == "T-ALL" | Type == "TEL-AML1" | Ty
pe == "Hyperdip50")
dis = dist(leukemia_subset, method="euclidean")
```

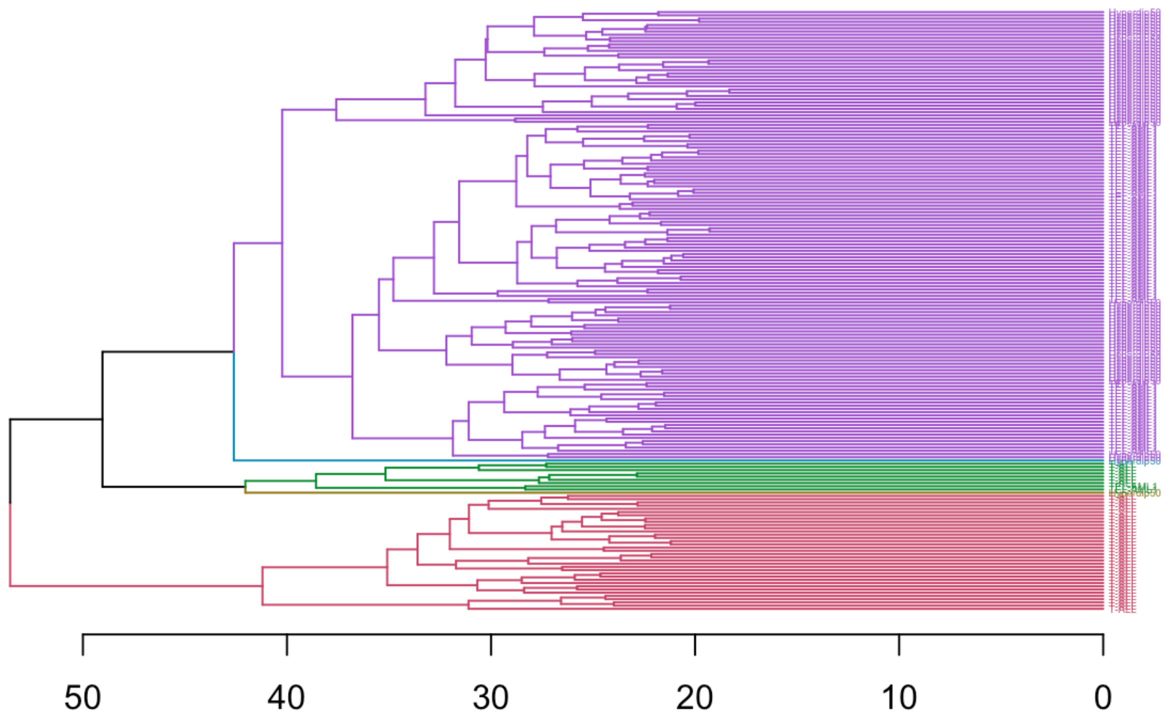
```
## Warning in dist(leukemia_subset, method = "euclidean"): NAs introduced by
## coercion
```

```
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")
```



```
dend1 = as.dendrogram(leukemia.hc)
dend1 = color_branches(dend1, k=5)
dend1 = color_labels(dend1, k=5)
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 5 clusters")
```

Dendrogram colored by 5 clusters


50
40
30
20
10
0