oblig2

Sanders

4/7/2020

Problem 1.

Reading dataset

```
df <- read.csv("http://web.stanford.edu/~hastie/CASI_files/DATA/leukemia_big.csv",
               header = T,
               sep = ","
               )
N_gene_expressions = 7128
N_patiants = 72

ALL_patients = grepl("ALL", names(df))
AML_patients = grepl("AML", names(df))
```

a)

```
library(pls)
```

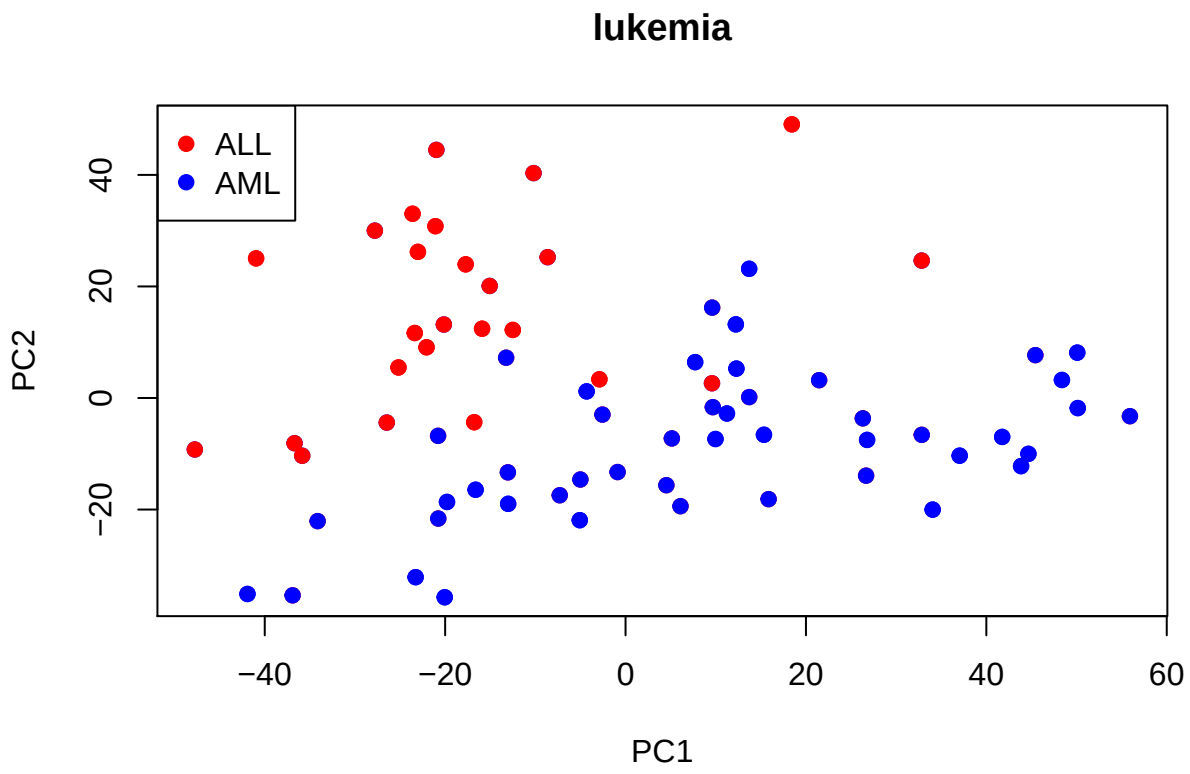
```
## Warning: package 'pls' was built under R version 3.6.3
```

```
##
## Attaching package: 'pls'
```

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

```
PC_analysis = prcomp(t(df), center = T, scale = T, rank. = 2)
plot(
  PC_analysis$x, col=c("blue", "red"), main="lukemia", xlab="PC1", ylab = "PC2", pch=1
)
X = PC_analysis$x

points(X[ALL_patients,1], X[ALL_patients, 2], col = "4", pch = 19)
points(X[AML_patients,1], X[AML_patients, 2], col = "2", pch = 19)
legend("topleft", legend = c("ALL", "AML"), col = c(2, 4), pch = c(19, 19))
```



b)

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.6.3
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 3.0-2
```

```
target_df <- read.csv("https://www.uio.no/studier/emner/matnat/math/STK2100/v20/eksamen/response_train.
               header = T,
               sep = ","
               )

# mod.Lasso.3 = glmnet(x = t(df), y = target_df[,2], alfa = 1, standardize = T, nfolds = 3)
lambda.l.a.3 = cv.glmnet(x =t(df), y = target_df[,2], alfa = 1, standardize = T, nfolds = 3)
lambda.l.a.3
```

```
##
## Call:  cv.glmnet(x = t(df), y = target_df[, 2], nfolds = 3, alfa = 1,      standardize = T)
##
## Measure: Mean-Squared Error
##
##     Lambda Measure    SE Nonzero
## min  1.231   81.77 10.04      36
## 1se  2.474   91.00 10.18      25
```

```
# mod.Lasso.10 = glmnet(x = t(df), y = target_df[,2], alfa = 1, standardize = T, nfolds = 10)
lambda.l.a.10 = cv.glmnet(x =t(df), y = target_df[,2], alfa = 1, standardize = T, nfolds = 10)
lambda.l.a.10
```

```
##
## Call:  cv.glmnet(x = t(df), y = target_df[, 2], nfolds = 10, alfa = 1,      standardize = T)
##
## Measure: Mean-Squared Error
##
##     Lambda Measure     SE Nonzero
## min 0.3195   71.89  9.866      66
## 1se 1.0221   81.15 11.065      44
```

```
# mod.Lasso.72 = glmnet(x = t(df), y = target_df[,2], alfa = 1, standardize = T, nfolds = 72)
lambda.l.a.72 = cv.glmnet(x =t(df), y = target_df[,2], alfa = 1, standardize = T, nfolds = 72)
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
## fold
```

```
lambda.l.a.72
```

```
##
## Call:  cv.glmnet(x = t(df), y = target_df[, 2], nfolds = 72, alfa = 1,      standardize = T)
##
## Measure: Mean-Squared Error
##
##     Lambda Measure    SE Nonzero
## min 0.3195   49.33 7.314      66
## 1se 0.9313   56.23 8.227      46
```

The penalty method used in lasso might not just help with **parameter restriction**, but also might requier some coeffisients to be zero. Effectivly reducing the complexity of the model. This is using a absolute value constraint. Where lambda (s) is a shrinking parameter where the absolute sum of the betas shuld be smaller then lambda.

c)

```
mod.Lasso.3 = glmnet(x = t(df), y = target_df[,2], alfa = 1, standardize = T, nfolds = 3, lambda = lambd
mod.Lasso.10 = glmnet(x = t(df), y = target_df[,2], alfa = 1, standardize = T, nfolds = 3, lambda = lamb
mod.Lasso.72 = glmnet(x = t(df), y = target_df[,2], alfa = 1, standardize = T, nfolds = 3, lambda = lamb

print("Non zero indexes for Lasso 3")
```

```
## [1] "Non zero indexes for Lasso 3"
```

```
print(which(mod.Lasso.3$beta != 0))
```

```
##  [1]    1   11   21   39   41   47   48   51   71  652  665 1090 1262 1484 2013
## [16] 2154 2168 2323 2552 4150 4629 4635 5038 5089 5183 5223 5470 5477 5603 5781
## [31] 5788 6165 6213 6518 6790 6889
```

```
print("Non zero indexes for Lasso 10")
```

```
## [1] "Non zero indexes for Lasso 10"
```

```
print(which(mod.Lasso.10$beta != 0))
```

```
##  [1]    1   11   21   39   41   51   71   81  138  393  464  689  887  909 1090
```

```
## [16] 1262 1304 1461 1484 2013 2154 2168 2323 2351 2461 2552 2966 3069 3240 3429
## [31] 3653 3696 4016 4058 4150 4432 4606 4629 4635 4671 4983 5038 5089 5183 5197
## [46] 5223 5470 5477 5508 5603 5781 5788 6165 6181 6213 6490 6518 6697 6771 6790
## [61] 6889 7052
```

```
print("Non zero indexes for Lasso 72")
```

```
## [1] "Non zero indexes for Lasso 72"
```

```
print(which(mod.Lasso.72$beta != 0))
```

```
##  [1]    1   11   21   39   41   51   71   81  138  393  464  689  887  909 1090
## [16] 1262 1304 1461 1484 2013 2154 2168 2323 2351 2461 2552 2966 3069 3240 3429
## [31] 3653 3696 4016 4058 4150 4432 4606 4629 4635 4671 4983 5038 5089 5183 5197
## [46] 5223 5470 5477 5508 5603 5781 5788 6165 6181 6213 6490 6518 6697 6771 6790
## [61] 6889 7052
```