Compulsory Exercise 2: Group 37

TMA4268 Statistical Learning V2019

Anders Bendiksen and Helge Bergo

20 March, 2020

Problem 1 (10p)

a) Ridge Regression (2p)

Show that the ridge regression estimator is $\hat{\beta}_{Ridge} = (X^T X + \lambda I)^{-1} X^T y$.

b) (2p)

Find the expected value and the variance-covariance matrix of $\hat{\beta}_{Ridge}$ (1P each).

c) (2P) - Multiple choice

(i) TRUE
(ii) FALSE
(iii) FALSE
(iv) TRUE

d) Forward Selection

```
library(ISLR)
set.seed(1)
train.ind = sample(1:nrow(College), 0.5*nrow(College))
college.train = College[train.ind,]
college.test = College[-train.ind,]
```

After dividing the data into a training and test set, the `regsubsets` function was used to create a forward selection model on the data, from the `leaps`-library.

```
library(leaps)
regfit.fwd = regsubsets(Outstate~.,data=college.train,method="forward", nvmax = 18)
reg.summary = summary(regfit.fwd)
```

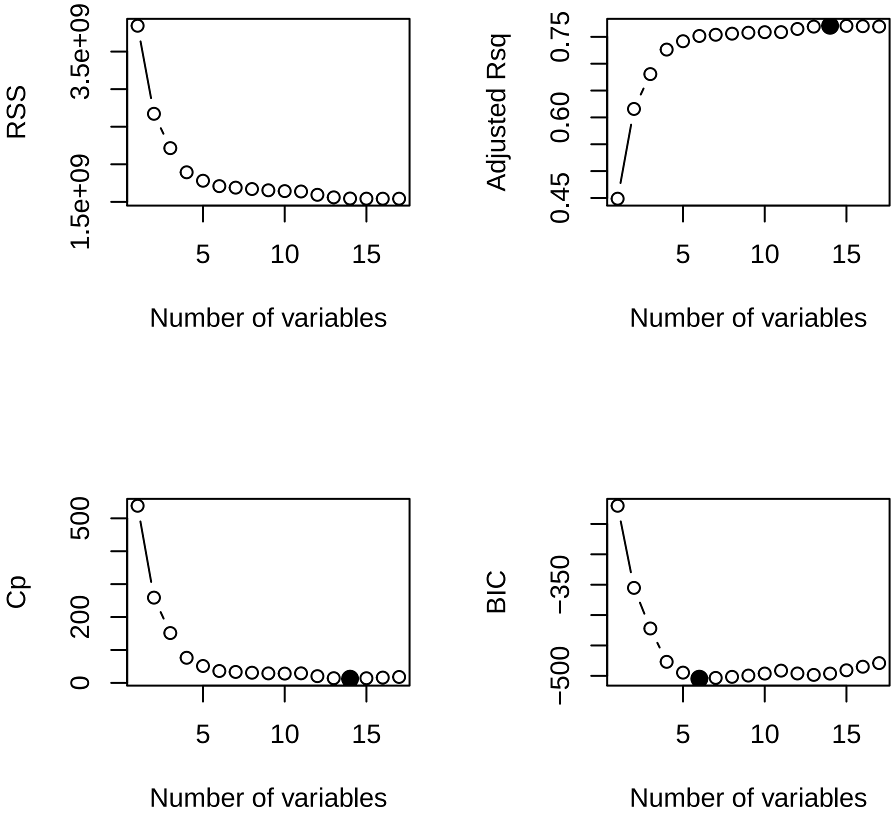
To decide on which model is best, the number of variables used in the selection was plotted against `RSS`, `Cp`, `BIC` and `adjusted R$^2$`.

```
par(mfrow=c(2,2))
plot(reg.summary$rss,xlab="Number of variables",ylab="RSS",type="b")
```

```r
plot(reg.summary$adjr2,xlab="Number of variables",ylab="Adjusted Rsq",type="b")
max.adjr2 = which.max(reg.summary$adjr2)
points(max.adjr2,reg.summary$adjr2[max.adjr2], col="black",cex=2,pch=20)

plot(reg.summary$cp,xlab="Number of variables",ylab="Cp",type="b")
min.cp = which.min(reg.summary$cp)
points(min.cp,reg.summary$cp[min.cp], col="black",cex=2,pch=20)

plot(reg.summary$bic,xlab="Number of variables",ylab="BIC",type="b")
min.bic = which.min(reg.summary$bic)
points(min.bic,reg.summary$bic[min.bic], col="black",cex=2,pch=20)
```



The maximum `adjusted` R^2 is the one with 14 variables, with a value of 0.7706887, shown as a filled dot in the upper right plot. This is also the same number of variables as for the lowest Cp. However, all the plots are pretty flat after around 6 or 7 variables used, and it seems like using only 6 variables still gives a good `adjusted` R^2 value of 0.7516133, without the increased complexity of adding 7 more variables. The model is then:

```r
coef(regfit.fwd,6)
```

```
##   (Intercept)    PrivateYes    Room.Board      Terminal   perc.alumni 
## -4726.8810613  2717.7019276     1.1032433    36.9990286    59.0863753 
##        Expend     Grad.Rate 
##     0.1930814    33.8303314
```

For the MSE, the following code calculates the MSE for all the variables.

```r
val.errors = rep(NA,17)
x.test = model.matrix(Outstate~.,data=college.test) # notice the -index!
```

```
for (i in 1:17) {
    coefi = coef(regfit.fwd,id=i)
    pred = x.test[,names(coefi)]%*%coefi
    val.errors[i] = mean((college.test$Outstate-pred)^2)
}

# plot(sqrt(val.errors),xlab="Number of variables", ylab="Root MSE",ylim=c(1500,5000) ,pch=19,type="b")
# points(sqrt(regfit.fwd$rss[-1]/180),col="blue",pch=19,type="b")
# legend("topright",legend=c("Training","Validation"),col=c("black","blue"),pch=19)
```

The MSE of the model with 6 variables is then:

```
val.errors[6]
```

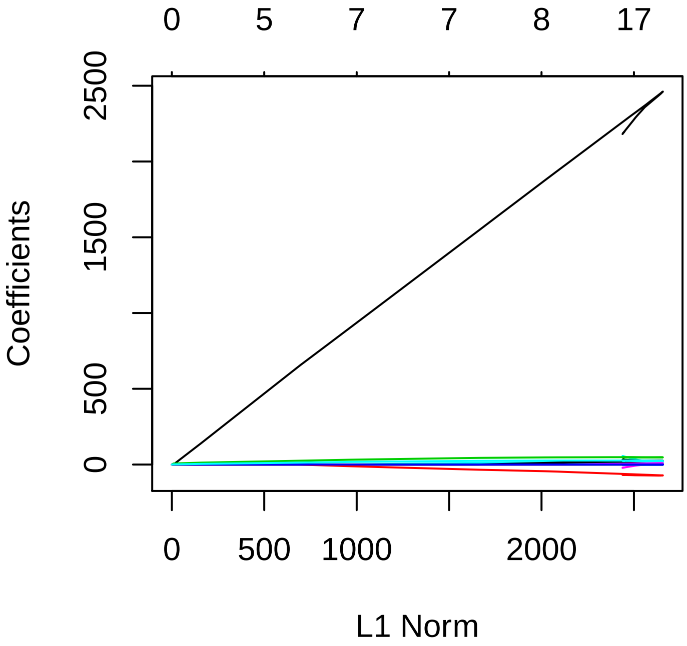
```
## [1] 3844857
```

e) (2p)

Using the Lasso method from the `glmnet`-library, a new model was selected.

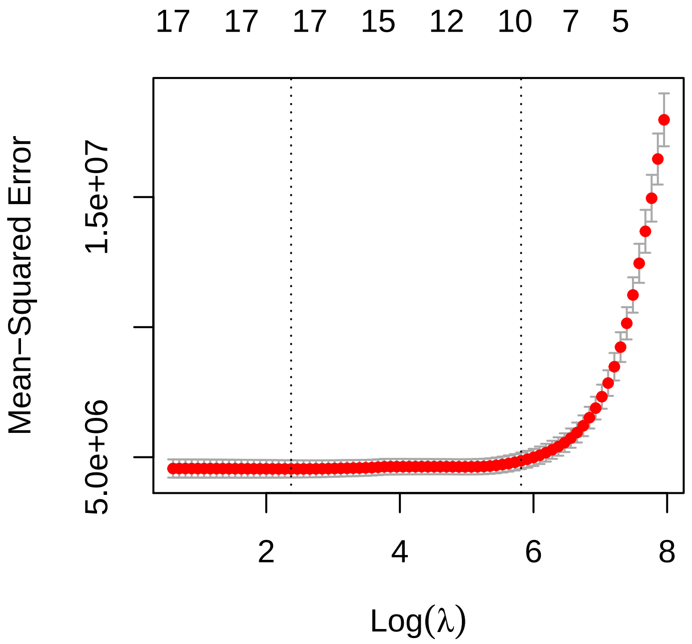
```
library(glmnet)
x.train = model.matrix(Outstate~.,data=college.train)[,-1]
y.train = college.train$Outstate
x.test = model.matrix(Outstate~.,data=college.test)[,-1]
y.test = college.test$Outstate

lasso.model = glmnet(x.train,y.train,alpha = 1)
plot(lasso.model)
```



To select the tuning parameter λ, cross-validation was performed, and the λ giving the lowest MSE was selected.

```
cv.out = cv.glmnet(x.train,y.train, alpha = 1)
plot(cv.out)
```



```
best.lambda = cv.out$lambda.min
best.lambda
```

```
## [1] 10.7207
```

This was used on the test set, to get the MSE for the

```
lasso.pred = predict(lasso.model,s=best.lambda ,newx=x.test)
MSE = mean((lasso.pred-y.test)^2)
MSE
```

```
## [1] 3688061
```

Finally, the coefficients of the model are shown here:

```
lasso.coef = predict(cv.out,type="coefficients",s=best.lambda)[1:18,]
lasso.coef
```

```
##   (Intercept)    PrivateYes          Apps        Accept        Enroll
## -1.172140e+03  2.230467e+03 -2.825215e-01  6.615811e-01 -3.778631e-01
##     Top10perc     Top25perc   F.Undergrad   P.Undergrad    Room.Board
##  4.589180e+01 -1.485674e+01 -5.800132e-02 -5.713770e-02  1.088115e+00
##         Books      Personal           PhD      Terminal     S.F.Ratio
## -9.185125e-01 -3.005419e-01  4.013410e+00  2.996744e+01 -6.936391e+01
##   perc.alumni        Expend     Grad.Rate
##  4.686967e+01  1.480013e-01  2.431539e+01
```

Problem 2 (9p)

a) (2p) - Multiple choice

Which of the following statements are true, which false?

(i) A regression spline of order 3 with 4 knots has 8 basis functions.

(ii) A regression spline with polynomials of degree $M-1$ has continous derivatives up to order $M-2$, but not at the knots.

(iii) A natural cubic spline is linear beyond the boundary knots.

(iv) A smoothing spline is (a shrunken version of) a natural cubic spline with knots at the values of all data points x_i for $i = 1, \dots, n$.

(v)

(vi)

(vii)

(viii)

b) (2p)

Write down the basis functions for a cubic spline with knots at the quartiles q_1, q_2, q_3 of variable X.

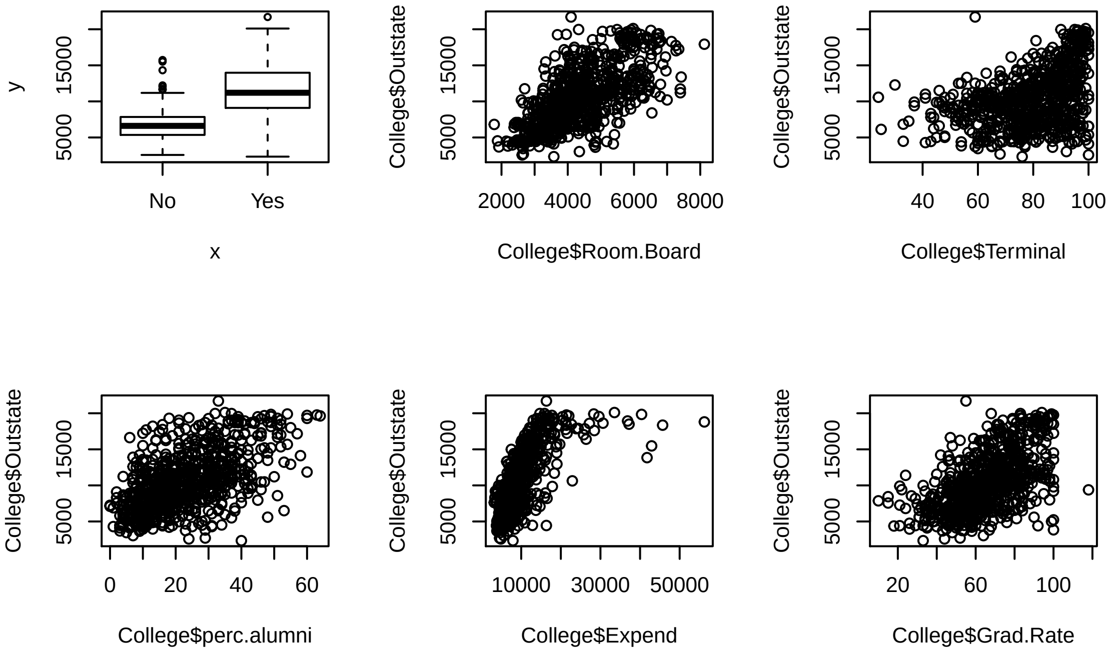
c) (2p)

We continue with using the `College` dataset that we used in problem 1. Investigate the relationships between `Outstate` and the following 6 predictors (using the training dataset `college.train`):

```
## [1] "Private"     "Room.Board"  "Terminal"    "perc.alumni" "Expend"
## [6] "Grad.Rate"
```

Create some informative plots and say which of the variables seem to have a linear relationship and which not and might thus benefit from a non-linear transformation (like e.g. a spline).

```r
par(mfrow=c(2,3))
plot(College$Private,College$Outstate)
plot(College$Room.Board,College$Outstate)
plot(College$Terminal,College$Outstate)
plot(College$perc.alumni,College$Outstate)
plot(College$Expend,College$Outstate)
plot(College$Grad.Rate,College$Outstate)
```



From these plots, it seems like `Room.board`, `perc.alumni` and `Grad.Rate` all have quite linear relationshps with `Outstate`, while both `Terminal` and `Expend` seem to follow a non-linear relationship.

d) (3P)

(i) Fit polynomial regression models for `Outstate` with `Terminal` as the only covariate for a range of polynomial degrees ($d = 1, \ldots, 10$) and plot the results. Use the training data (`college.train`) for this task.

```
for (i in 1:10) {
  poly.fit = lm(Outstate ~ poly(Terminal,i), data=college.train)
  # plot(poly.fit)
}
coef(summary(poly.fit))
```

```
##                          Estimate Std. Error    t value      Pr(>|t|)
## (Intercept)            10484.0000   191.4776 54.7531395 1.592465e-181
## poly(Terminal, i)1     33775.7920  3771.6714  8.9551258  1.576072e-17
## poly(Terminal, i)2     16996.9239  3771.6714  4.5064700  8.807400e-06
## poly(Terminal, i)3      5610.8187  3771.6714  1.4876213  1.376867e-01
## poly(Terminal, i)4       906.6817  3771.6714  0.2403925  8.101566e-01
## poly(Terminal, i)5     -2479.4660  3771.6714 -0.6573918  5.113301e-01
## poly(Terminal, i)6       651.1277  3771.6714  0.1726364  8.630299e-01
## poly(Terminal, i)7     -5472.4748  3771.6714 -1.4509416  1.476277e-01
## poly(Terminal, i)8     -4631.9185  3771.6714 -1.2280811  2.201827e-01
## poly(Terminal, i)9     -9973.8438  3771.6714 -2.6444095  8.525278e-03
## poly(Terminal, i)10 -2737.9707  3771.6714 -0.7259303  4.683319e-01
```

```
library(ISLR)
# extract only the two variables from Auto
ds = Auto[c("horsepower", "mpg")]
ds = college.train[c("Terminal","Outstate")]
n = nrow(ds)
```

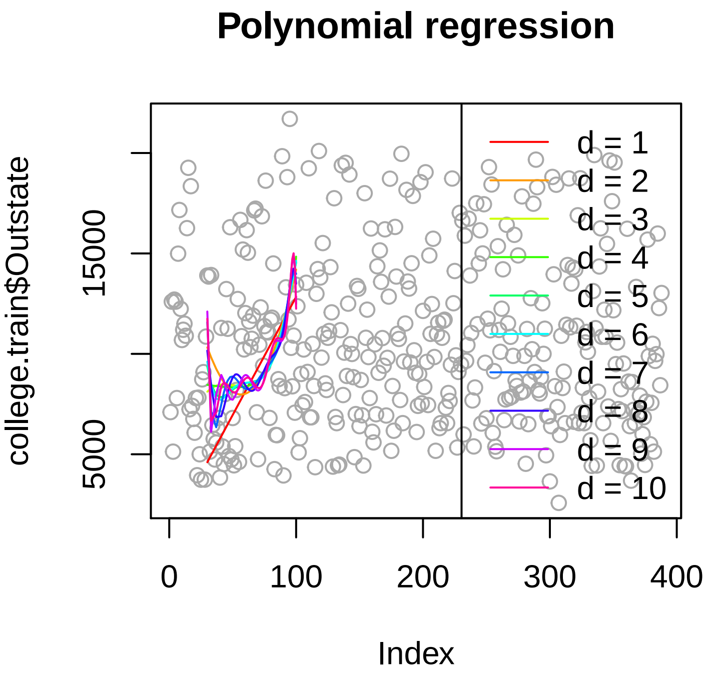
```r
# which degrees we will look at
deg = 1:10

# training ids for training set
tr = sample.int(n, n/2)
# plot of training data
# plot(ds[tr, ], col = "darkgrey", main = "Polynomial regression")
plot(college.train$Outstate,col = "darkgrey", main = "Polynomial regression")

# which colors we will plot the lines with
colors = rainbow(length(deg))
# iterate over all degrees (1:4) - could also use a for-loop here
MSE = sapply(deg, function(d) {
  # fit model with this degree
  model = lm(Outstate ~ poly(Terminal, d), data=college.train)

  lines(cbind(college.train$Terminal, model$fit)[order(college.train$Terminal) ,], col = colors[d])
  # add lines to the plot - use fitted values (for mpg) and horsepower from # training set
  # lines(cbind(ds[tr, 1], model$fit)[order(ds[tr, 1]), ], col = colors[d])

  # calculate mean MSE - this is returned in the MSE variable
  # mean((predict(mod, ds[-tr, ]) - ds[-tr, 2])^2)
})
# add legend to see which color corresponds to which line
legend("topright", legend = paste("d =", deg), lty = 1, col = colors)
```



(ii) Still for the training data, choose a suitable smoothing spline model to predict `Outstate` as a function of `Expend` (again as the only covariate) and plot the fitted function into the scatterplot of `Outstate` against `Expend`. How did you choose the degrees of freedom?

(iii) Report the corresponding training MSE for (i) and (ii). Did you expect that?

Problem 3 (9p)

a) (2P) - Multiple choice

Which of the following statements are true, which false?

(i) Regression trees cannot handle categorical predictors.
(ii) Regression and classification trees are easy to interpret.
(iii) The random forest approaches improves bagging, because it reduces the variance of the predictor function by decorrelating the trees.
(iv) The number of trees B in bagging and random forests is a tuning parameter.

b) (4P)

Select one method from Module 8 (tree-based methods) in order to build a good model to predict `Outstate` in the `College` dataset that we used in problems 1 and 2. Explain your choice (pros/cons?) and how you chose the tuning parameter(s). Train the model using the training data and report the MSE for the test data.

c) (2p)

Compare the results (tests MSEs) among all the methods you used in Problems 1-3. Which method perform best in terms of prediction error? Which method would you choose if the aim is to develop an interpretable model?

Problem 4 (12P)

We will use the classical data set of *diabetes* from a population of women of Pima Indian heritage in the US, available in the R `MASS` package. The following information is available for each woman:

- diabetes: `0`= not present, `1`= present
- npreg: number of pregnancies
- glu: plasma glucose concentration in an oral glucose tolerance test
- bp: diastolic blood pressure (mmHg)
- skin: triceps skin fold thickness (mm)
- bmi: body mass index (weight in kg/(height in m)^2)
- ped: diabetes pedigree function.
- age: age in years

We will use a training set (called `d.train`) with 300 observations (200 non-diabetes and 100 diabetes cases) and a test set (called `d.test`) with 232 observations (155 non-diabetes and 77 diabetes cases). Our aim is to make a classification rule for the presence of diabetes (yes/no) based on the available data. You can load the data as follows:

```
id <- "1Fv6xwKLSZHldRAC1MrcK2mzdOYnbgv0E" # google file ID
d.diabetes <- dget(sprintf("https://docs.google.com/uc?id=%s&export=download", id))
d.train=d.diabetes$ctrain
d.test=d.diabetes$ctest
```

a) (2P) - Multiple choice

Start by getting to know the *training data*, by producing summaries and plots. Which of the following statements are true, which false?

(i) Females with high glucose levels and higher bmi seem to have a higher risk for diabetes.
(ii) Some women had up to 17 pregnancies.
(iii) BMI and triceps skin fold thickness seem to be positively correlated.
(iv) The distribution of the number of pregnancies per woman seems to be a bit skewed and a transformation of this variable could therefore be appropriate.

b) (4P)

Fit a support vector classifier (linear boundary) and a support vector machine (radial boundary) to find good functions that predict the diabetes status of a patient. Use cross-validation to find a good cost parameter (for the linear boundary) and a good combination of cost *and* γ parameters (for the radial boundary). Report the confusion tables and misclassification error rates for the test set in both cases. Which classifier do you prefer and why? (Do not use any variable transformations or standardizations to facilitate correction).

R-hints: The response variable must be converted into a factor variable before you continue.

```r
d.train$diabetes <- as.factor(d.train$diabetes)
d.test$diabetes <- as.factor(d.test$diabetes)
```

To run cross-validation over a grid of two tuning parameters, you can use the `tune()` function where `ranges` defines the grid points as follows:

```r
tune(..., formula, kernel=...,ranges=list(cost=c(...), gamma=c(...)))
```

c) (2P)

Compare the performance of the two classifiers from b) to *one other classification method* that you have learned about in the course. Explain your choice and report the confusion table and misclassification error rate on the test set for your chosen method and interpret what you see. What are advantages/disadvantages of your chosen method with respect to SVMs?

d) (2P) - Multiple choice

Which of the following statements are true, which false?

(i) Under standard conditions, the maximal margin hyperplane approach is equivalent to a linear discriminant analysis.
(ii) Under standard conditions, the support vector classifier is equivalent to quadratic discriminant analysis.
(iii) Logistic regression, LDA and support vector machines tend to perform similar when decision boundaries are linear, unless classes are linearly separable.
(iv) An advantage of logistic regression over SVMs is that it is easier to do feature selection and to interpret the results.

e) (2P) Link to logistic regression and hinge loss.

Look at slides 71-73 of Module 9. Show that the loss function

$$\log(1 + \exp(-y_i f(\boldsymbol{x}_i)))$$

is the deviance for the $y = -1, 1$ encoding in a logistic regression model.
Hint: $f(\boldsymbol{x}_i)$ correponds to the linear predictor in the logistic regression approach.

Problem 5 (10P)

The following dataset consists of 40 tissue samples with measurements of 1,000 genes. The first 20 tissues come from healthy patients and the remaining 20 come from a diseased patient group. The following code loads the dataset into your session with row names decribing if the tissue comes from a diseased or healthy person.

```
# id <- "1VfVCQvWt121UN39NXZ4aR9Dmsbj-p9OU" # google file ID
# GeneData <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download", id),header=F)
# colnames(GeneData)[1:20] = paste(rep("H", 20), c(1:20), sep = "")
# colnames(GeneData)[21:40] = paste(rep("D", 20), c(1:20), sep = "")
# # row.names(GeneData) = paste(rep("G", 1000), c(1:1000), sep = "")
```

a) (2P)

Perform hierarchical clustering with complete, single and average linkage using **both** Euclidean distance and correlation-based distance on the dataset. Plot the dendograms. Hint: You can use `par(mfrow=c(1,3))` to plot all three dendograms on one line or `par(mfrow=c(2,3))` to plot all six together.

b) (2P)

Use these dendograms to cluster the tissues into two groups. Compare the groups with respect to the patient group the tissue comes from. Which linkage and distance measure performs best when we know the true state of the tissue?

c) (1P)

With Principal Component Analysis, the first principal component loading vector solves the following optimization problem,

$$\max_{\phi_{11},\ldots,\phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} \phi_{j1}^2 = 1.$$

Explain what ϕ, p, n and x are in this optimization probelm and write down the formula for the first principal component scores.

d) (2P)

(i) (1P) Use PCA to plot the samples in two dimensions. Color the samples based on the tissues group of patients.
(ii) (1P) How much variance is explained by the first 5 PCs?

e) (1P)

Use your results from PCA to find which genes that vary the most accross the two groups.

f) (2P)

Use K-means to seperate the tissue samples into two groups. Plot the values in a two-dimensional space with PCA. What is the error rate of K-means?