Exercise 8

Tobias Famos

Preliminaries

Read the data

```
vertebral <- read.csv("/home/tobias/unibe/statistical methods in R/Exercise 8/Vertebral.txt", as.is=F)
summary(vertebral)
```

```
##    Incidence          Tilt            Angle            Slope
##  Min.   : 26.15   Min.   :-6.555   Min.   : 14.00   Min.   : 13.37
##  1st Qu.: 46.43   1st Qu.:10.667   1st Qu.: 37.00   1st Qu.: 33.35
##  Median : 58.69   Median :16.358   Median : 49.56   Median : 42.40
##  Mean   : 60.50   Mean   :17.543   Mean   : 51.93   Mean   : 42.95
##  3rd Qu.: 72.88   3rd Qu.:22.120   3rd Qu.: 63.00   3rd Qu.: 52.70
##  Max.   :129.83   Max.   :49.432   Max.   :125.74   Max.   :121.43
##      Radius           Degree              Status
##  Min.   : 70.08   Min.   :-11.058   Abnormal:210
##  1st Qu.:110.71   1st Qu.:  1.604   Normal  :100
##  Median :118.27   Median : 11.768
##  Mean   :117.92   Mean   : 26.297
##  3rd Qu.:125.47   3rd Qu.: 41.287
##  Max.   :163.07   Max.   :418.543
```

```
library(boot)
```

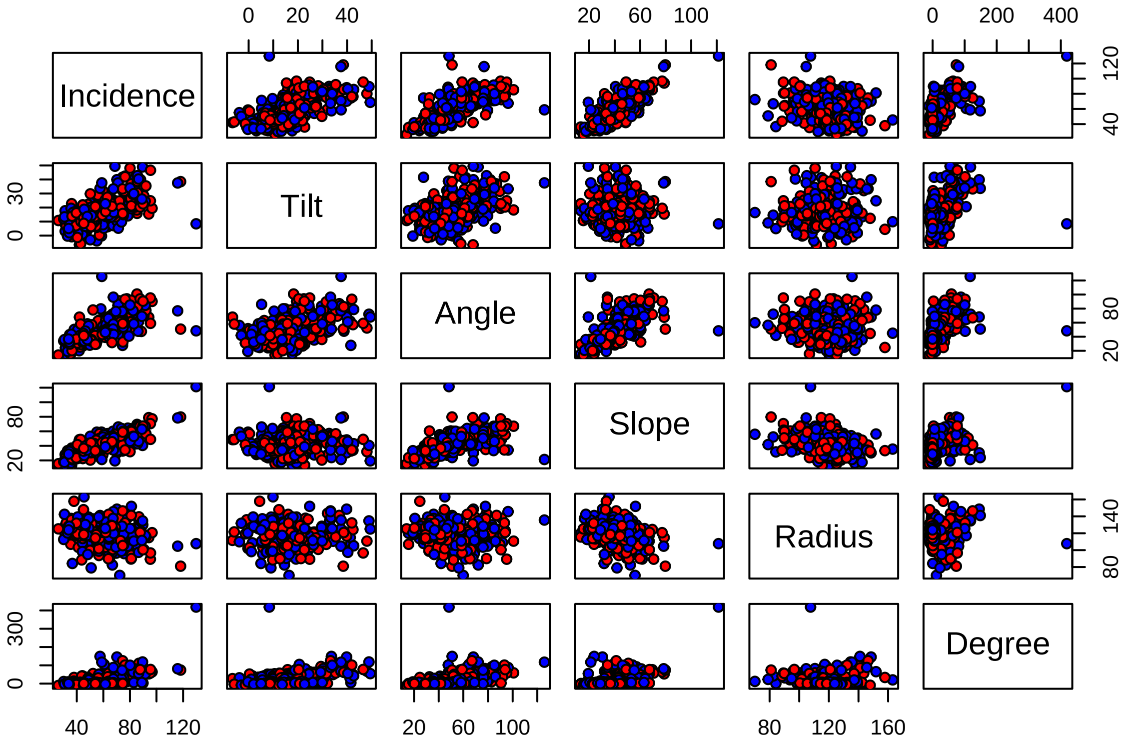
Check for NA / missing values

```
sum(is.na(vertebral))
```

```
## [1] 0
```

Take a look at the data

```
only_predictors <- subset(vertebral, select=-Status)
pairs(only_predictors, bg=c("red", "blue"), pch=21)
```



Import the library for LDA and cross validation

```
library(MASS)
library(boot)
```

Logistic Regression

Build an initial logistic regression modell

```
vertebral.logistic <- glm(Status~., data=vertebral, family=binomial)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(vertebral.logistic)
```

```
##
## Call:
## glm(formula = Status ~ ., family = binomial, data = vertebral)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2678  -0.3639  -0.0289   0.4081   2.7317
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.530e+01  3.315e+00  -4.615 3.93e-06 ***
## Incidence    2.517e+07  4.017e+07   0.627    0.531
## Tilt        -2.517e+07  4.017e+07  -0.627    0.531
## Angle        1.794e-02  2.290e-02   0.784    0.433
## Slope       -2.517e+07  4.017e+07  -0.627    0.531
## Radius       1.077e-01  2.318e-02   4.645 3.39e-06 ***
## Degree      -1.693e-01  2.335e-02  -7.248 4.23e-13 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 389.86  on 309  degrees of freedom
## Residual deviance: 177.87  on 303  degrees of freedom
## AIC: 191.87
##
## Number of Fisher Scoring iterations: 8
```

Build again without the not significant predictors Incidence, Tilt, Angle and Slope

```
vertebral.logistic <- glm(Status~Radius + Degree, data=vertebral, family=binomial)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(vertebral.logistic)
```

```
##
## Call:
## glm(formula = Status ~ Radius + Degree, family = binomial, data = vertebral)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.09309  -0.35501  -0.06575   0.69722   2.52359
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.32320    2.06244  -4.036 5.45e-05 ***
## Radius       0.07414    0.01712   4.330 1.49e-05 ***
## Degree      -0.11022    0.01761  -6.258 3.90e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 389.86  on 309  degrees of freedom
## Residual deviance: 219.66  on 307  degrees of freedom
## AIC: 225.66
##
## Number of Fisher Scoring iterations: 7
```

```
vertebral.logistic.cv <- cv.glm( vertebral,vertebral.logistic, K=10)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
vertebral.logistic.cv $delta[1]
```

```
## [1] 0.1239022
```

Thus we derive at an Error rage of 1 - 0.12344 = **0.87656** ## LDA Build the Linear Discriminant Analysis

```
cor(only_predictors)
```

```
##            Incidence       Tilt       Angle       Slope      Radius      Degree
## Incidence  1.0000000 0.62919877  0.71728236  0.81495999 -0.24746721  0.63874275
## Tilt       0.6291988 1.00000000  0.43276386  0.06234529  0.03266781  0.39786228
## Angle      0.7172824 0.43276386  1.00000000  0.59838689 -0.08034361  0.53366701
## Slope      0.8149600 0.06234529  0.59838689  1.00000000 -0.34212835  0.52355746
## Radius    -0.2474672 0.03266781 -0.08034361 -0.34212835  1.00000000 -0.02606501
## Degree     0.6387427 0.39786228  0.53366701  0.52355746 -0.02606501  1.00000000
```

```
vertebral.lda <- lda(Status ~ Slope + Tilt+ Angle+Radius+Degree, data=vertebral, CV = TRUE)
table <- table(vertebral$Status, vertebral.lda$class, dnn = c('Actual Group','Predicted Group'))
table
```

```
##             Predicted Group
## Actual Group Abnormal Normal
##     Abnormal      191     19
##     Normal         27     73
```

```
1 - (table[1,1] + table[2,2])/length(vertebral$Status)
```

```
## [1] 0.1483871
```

As we can see, the accuracy lies at (191+73)/310 = **0.8516**. As this accuracy is derived at via cross validation we can take it as more reliable as if we would have just derived it from a reevaluation with the train data.

Task 3 Assessing

To assess a model fairly we must test it with new data (test data). Thus we need first to split the data in a train and test data set. This might reduce the accuracy of the models we build (depending on the model more or less) but it helps us to evaluate the models better and should not be ignored.

The methodology can be seen as fair, as both models get the same train and test data and thus have equality of oppurtunity. But, one could also argue that a smaller train set is an advantage for the LDA, as it generally performs better with small sets than the logistic regression.only

```
set.seed(2022)
sample <- sample(c(TRUE, FALSE), nrow(vertebral), replace=TRUE, prob=c(0.7,0.3))
train  <- vertebral[sample, ]
test   <- vertebral[!sample, ]
```

Build the LDA model (note that we need to set cross validation to false, as this is needed for making predictions with the model on new data.

```
train.lda <- lda(Status ~ Slope + Tilt+ Angle+Radius+Degree, data=vertebral, CV = FALSE)
predictions <- predict(train.lda, test)
prediction_table <- table(test$Status, predictions$class, dnn = c('Actual Group','Predicted Group'))
```

```r
error_rate.lda <- 1-(prediction_table[1,1] + prediction_table[2,2])/length(test$Status)
error_rate.lda
```

```
## [1] 0.1313131
```

Build the logistic regression model with the train data and test it with the new test data.

```r
train.log_reg <-  glm(Status~Radius + Degree, data=vertebral, family=binomial)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
predictions <- predict(train.log_reg, test, type="response")
```

Turn the probabilities into Abnormal and Normal and calculate the error Rate

```r
predictions[which(predictions<0.5)]<- "Abnormal"
predictions[which(predictions!="Abnormal")]<- "Normal"
prediction_table.log_reg <- table(test$Status, predictions, dnn = c('Actual Group','Predicted Group'))
prediction_table.log_reg
```

```
##              Predicted Group
## Actual Group Abnormal Normal
##     Abnormal       59     11
##     Normal          5     24
```

```r
error_rate.log_reg <- 1-(prediction_table.log_reg[1,1] + prediction_table.log_reg[2,2])/length(test$Sta
error_rate.log_reg
```

```
## [1] 0.1616162
```

We arrive at the following error Rates

Model	Error Rate Train / Test split	Error Rate Resubstitution
LDA	0.1313131	0.1483871
Logistic Regression	0.1616162	0.1220684

Thus we can conclude that based on a 30 / 70 test train split the Logistic Regression has a slightly higher error rate than the Linear Discriminative Analysis. In the resubstitution method with cross validation the Logistic Regression has a slightly lower error rate as the LDA.

As the resubstitution method does not punish overfitting and even encourages it, i would still tend to state that the LDA model is the better predictor model for this dataset.