Machine Learning for Biostatistics

Module 2

Armando Teixeira-Pinto

2025-07-18

Contents

Regression and Classification

Introduction

This module will cover traditional methods for prediction of continuous outcomes and categorical ones. In machine learning, these methods are known as regression (for continuous outcomes) and classification (for categorical outcomes) methods. This sometimes is a bit confusing given that, despite its name, *logistic* regression is a classification method under this terminology because in statistics, regression is used to refer to many models associating any type of outcome with independent variables.

In this module we are going to review the linear regression and describe the k-nearest neighbour regression. For the classification methods, we will explore three widely-used classifiers: logistic regression, K-nearest neighbours and linear discriminant analysis.

By the end of this module you should be able to:

1. Use linear regression for prediction
2. Estimate the *mean squared error* of a predictive model
3. Use knn regression and knn classifier
4. Use logistic regression as a classification algorithm
5. Calculate the confusion matrix and evaluate the classification ability
6. Implement linear and quadratic discriminant analyses

Datasets used in the examples

The file bmd.csv contains 169 records of bone densitometries (measurement of bone mineral density). The following variables were collected:

- id – patient's number
- age – patient's age
- fracture – hip fracture (fracture / no fracture)

- weight_kg – weight measured in Kg
- height_cm – height measure in cm
- waiting_time – time the patient had to wait for the densitometry (in minutes)
- bmd – bone mineral density measure in the hip

The file SBI.csv contains 2349 records of children admitted to the emergency room with fever and tested for serious bacterial infection (**sbi**). The following variables are included :

- id – patient's number
- fever_hours – duration of the fever in hours
- age – child's age
- sex – child's sex (M / F)
- wcc – white cell count
- prevAB – previous antibiotics (Yes / No)
- sbi – serious bacterial infection (Not Applicable / UTI / Pneum / Bact)
- pct – procalcitonin
- crp – c-reactive protein

The dataset bdiag.csv contains quantitative information from digitized images of a diagnostic test (fine needle aspirate (FNA) test on breast mass) for the diagnosis of breast cancer. The variables describe characteristics of the cell nuclei present in the image.

Variables Information:

- ID number
- Diagnosis (M = malignant, B = benign)

and ten real-valued features are computed for each cell nucleus:

- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness (perimeter^2 / area - 1.0)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

The mean, standard error and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

This database is also available through the UW CS ftp server: ftp ftp.cs.wisc.edu cd math-prog/cpo-dataset/machine-learn/WDBC/

Slides from the videos

You can download the slides used in the videos for Regression and Classification:

Slides

Chapter 1

Linear Regression

1.1 Introduction

You should be familiar with linear regression, so this section is likely a review of this model. Also, linear regression is a well established method and it is well studied, both from the theoretical and practical perspective. Therefore, there are many aspects that are referred in the textbook but we will not explore much in this section, such as, outliers, testing, heteroscedasticity, leverage power, but you should be familiar with these terms.

1.2 Readings

Read the following chapters of *An introduction to statistical learning*:

- 3.1 Simple Linear Regression
- 3.2 Multiple Linear Regression
- 3.3 Other Considerations in the Regression Model

1.3 Practice session

Task 1 - Fit a linear model

With the bmd.csv dataset, we want to fit a linear model to predict bone mineral density (BMD) based on AGE, SEX and BMI (BMI has to be computed) and we want to compute the R^2 and MSE for the models that were fitted.

Let's first read the data and compute "BMI"

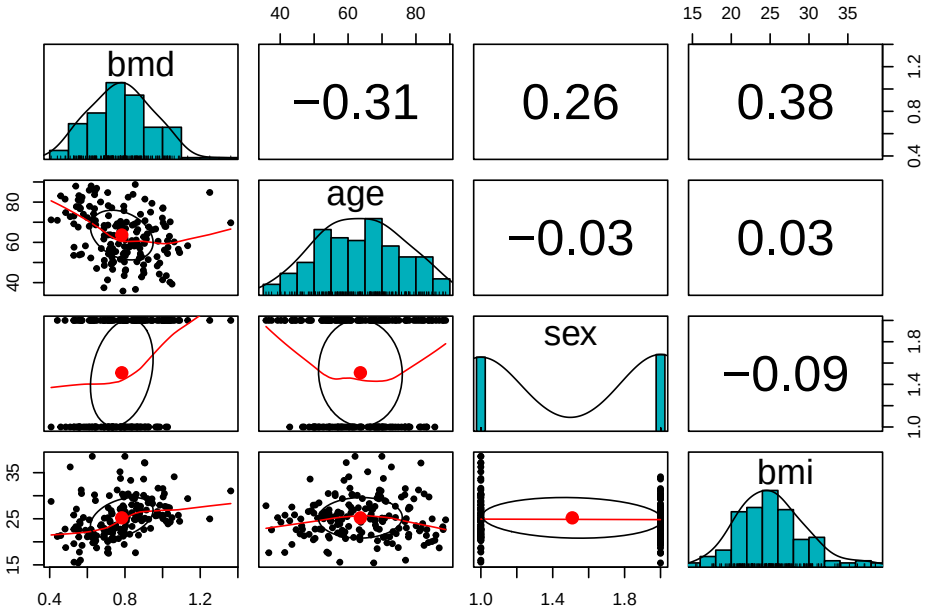
```
#libraries that we will need
library(psych) #for the function pairs.panels
set.seed(1974) #fix the random generator seed
```

```
#read the dataset
bmd.data     <-
  read.csv("https://www.dropbox.com/s/c6mhgatkotuze8o/bmd.csv?dl=1",
           stringsAsFactors = TRUE)

bmd.data$bmi <- bmd.data$weight_kg / (bmd.data$height_cm/100)^2
summary(bmd.data)
```

```
##        id              age        sex           fracture     weight_kg
##  Min.   :   35   Min.   :35.81   F:83   fracture   : 50   Min.   :36.00
##  1st Qu.: 2018   1st Qu.:54.42   M:86   no fracture:119   1st Qu.:56.00
##  Median : 6702   Median :63.49                            Median :64.50
##  Mean   : 9103   Mean   :63.63                            Mean   :64.67
##  3rd Qu.:17100   3rd Qu.:72.08                            3rd Qu.:73.00
##  Max.   :24208   Max.   :88.75                            Max.   :96.00
##    height_cm               medication   waiting_time        bmd
##  Min.   :142.0   Anticonvulsant :  9   Min.   : 5.00   Min.   :0.4076
##  1st Qu.:154.0   Glucocorticoids: 24   1st Qu.: 9.00   1st Qu.:0.6708
##  Median :160.5   No medication  :136   Median :14.00   Median :0.7861
##  Mean   :160.3                         Mean   :19.74   Mean   :0.7831
##  3rd Qu.:166.0                         3rd Qu.:24.00   3rd Qu.:0.8888
##  Max.   :177.0                         Max.   :96.00   Max.   :1.3624
##       bmi
##  Min.   :15.43
##  1st Qu.:22.15
##  Median :24.96
##  Mean   :25.20
##  3rd Qu.:27.55
##  Max.   :38.54
```

Before we model, let's look at the correlation structure of the variables involved

```
pairs.panels(bmd.data[c("bmd", "age","sex", "bmi")],
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
             )
```



We fit a linear model for BMD and evaluate the R-squared

```
#Fits a linear model with fixed effects only
  model1.bmd <- lm(bmd ~ age + sex + bmi, data = bmd.data)
  summary(model1.bmd)
```

```
##
## Call:
## lm(formula = bmd ~ age + sex + bmi, data = bmd.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38207 -0.07669 -0.00654  0.07888  0.51256
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.6063945  0.0834051   7.270 1.36e-11 ***
## age         -0.0041579  0.0008625  -4.821 3.23e-06 ***
## sexM         0.0949602  0.0213314   4.452 1.56e-05 ***
## bmi          0.0155913  0.0024239   6.432 1.30e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.138 on 165 degrees of freedom
## Multiple R-squared:  0.3254, Adjusted R-squared:  0.3131
## F-statistic: 26.53 on 3 and 165 DF,  p-value: 4.677e-14
```

```
  mean(model1.bmd$residuals^2) #MSE
```

```
## [1] 0.01859697
```

TRY IT YOURSELF:

1) Fit a linear model with the interaction age*sex - call it model 2

See the solution code

```
#Fits a linear model with an interaction age*sex
  model2.bmd <- lm(bmd ~ age * sex + bmi, data = bmd.data)
  summary(model2.bmd)
  mean(model2.bmd$residuals^2) #MSE
```

2. Fit a linear model with the the interaction age*sex and cubic effect for BMI - call it model 3

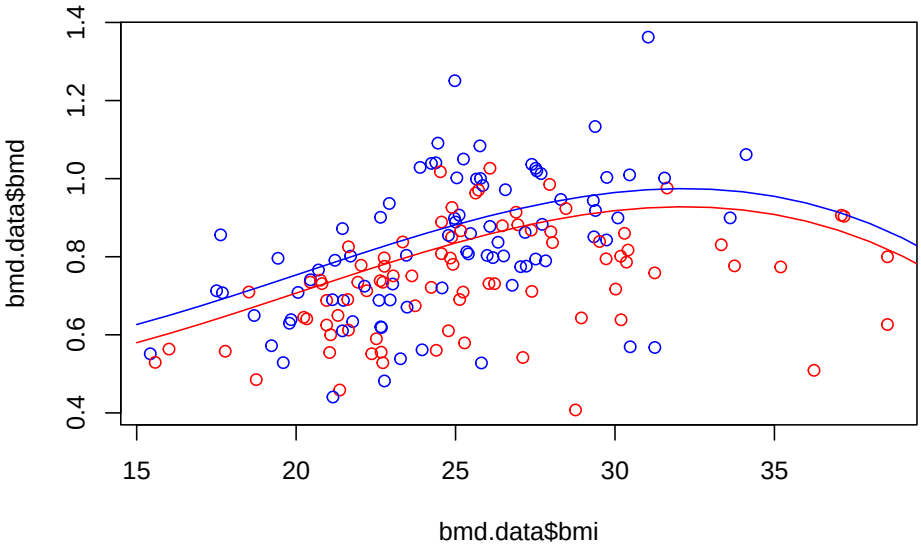
See the solution code

```
#Fits a linear model with an interaction and polynomial f
  model3.bmd <- lm(bmd ~ age*sex  + bmi + I(bmi^2) + I(bmi^3),
                   data = bmd.data)

  #You could use the poly() function to fit the same model
  #however, poly() will use orthogonal polynomials
  #so the coefficients will not be the same as above
  #summary(lm(bmd ~ age*sex  + poly(bmi,3) , data = bmd.data))
  summary(model3.bmd)
  mean(model3.bmd$residuals^2) #MSE
```

Task 2 - Predicting from a linear model

We first plot the scatter for BMD and BMI, then get the predictions from model 3 in task 1, for a new data where age=50, sex=F and we let BMI vary from 15 to 40. We also compute the predictions for males with similar characteristics. Finally, we add the fitted lines to the plot.

```
#Scatter plot of BMD and BMI
  plot(bmd.data$bmi, bmd.data$bmd,
       col = ifelse(bmd.data$sex=="F", "red", "blue"))
#prediction from model b) in task 1
  bmd.f50 <- predict(model3.bmd,
                   newdata = data.frame(age=50, sex="F", bmi=seq(15,40)))
  bmd.m50 <- predict(model3.bmd,
                   newdata = data.frame(age=50, sex="M", bmi=seq(15,40)))
  lines(seq(15,40), bmd.f50, col="red")
  lines(seq(15,40), bmd.m50, col="blue")
```

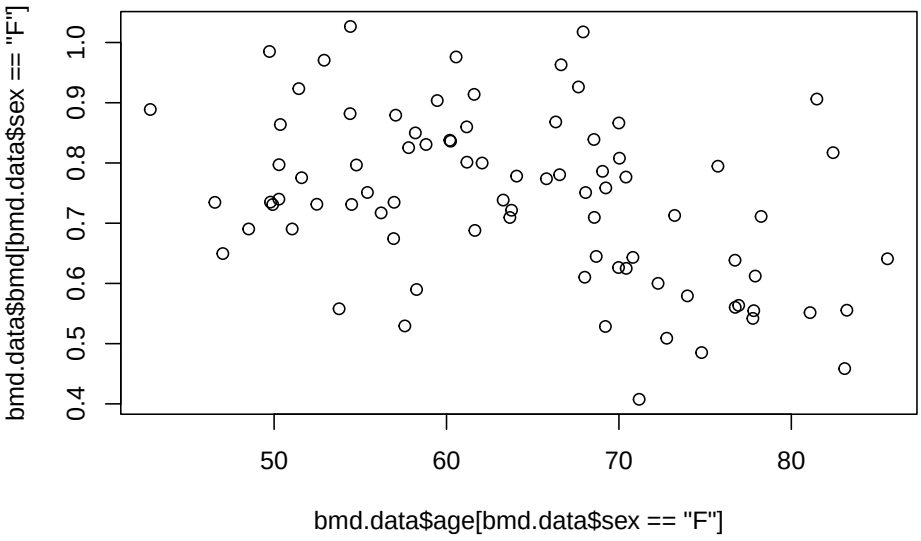


TRY IT YOURSELF:

1) Produce the scatter plot for BMD and AGE, only for women

See the solution code

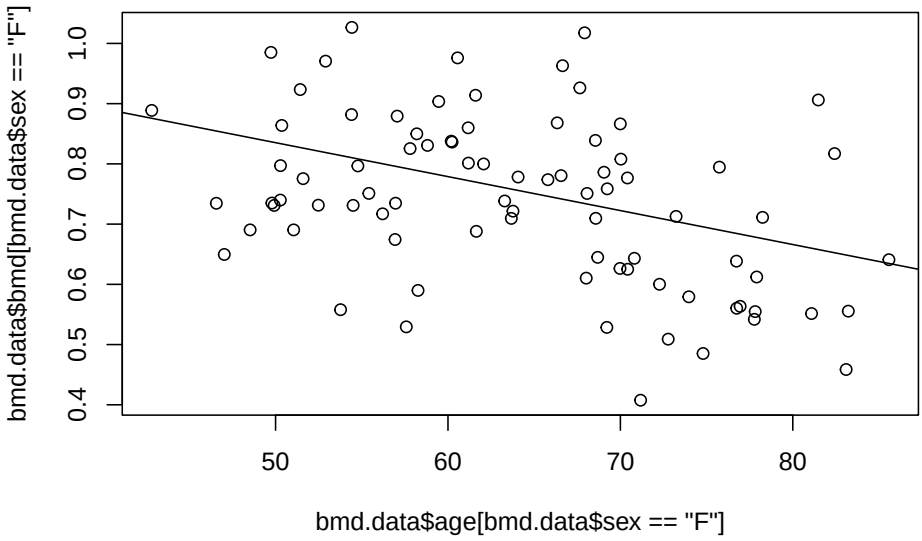
```
#Scatter plot of BMD and AGE
  plot( bmd.data$age[bmd.data$sex=="F"], bmd.data$bmd[bmd.data$sex=="F"])
```



2) Predict the BMD for women, with a BMI=25 and AGE between 40 and 90, using model 3 from task 1 and plot the prediction

See the solution code

```r
#Scatter plot of BMD and AGE
  plot( bmd.data$age[bmd.data$sex=="F"], bmd.data$bmd[bmd.data$sex=="F"])

#prediction from model 3 in task 1
#(the prediction line only )
  bmd.bmi25 <- predict(model3.bmd,
                       newdata = data.frame(age=seq(40,90), sex="F", bmi=25))
  lines(seq(40,90), bmd.bmi25)
```



1.4 Exercises

Solve the following exercises from the *An introduction to statistical learning* book:

1) Exercise 4 (page 122)

2) Exercise 13 (page 126)

3) With the *fat* dataset in the `library(faraway)`, we want to fit a linear model to predict body fat (variable **brozek**) using the variable **abdom** and **age**. After loading the library, use the command `data(fat)` to load the dataset.

   a) recode the variable age into **age_cat** with the following categories: <30, 30-50 and >50

   b) fit a linear model using **abdom** and **age_cat** and compute the mean squared error

   c) fit a linear model using **abdom** , **age_cat** and the interaction between

these two predictors. Comment on the change in the mean squared error of this model compared to the one without the interaction.

Chapter 2

K-nearest Neighbours Regression

2.1 Introduction

KNN regression is a non-parametric method that, in an intuitive manner, approximates the association between independent variables and the continuous outcome by averaging the observations in the same *neighbourhood*. The size of the neighbourhood needs to be set by the analyst or can be chosen using cross-validation (we will see this later) to select the size that minimises the mean-squared error.

While the method is quite appealing, it quickly becomes impractical when the dimension increases, i.e., when there are many independent variables.

2.2 Readings

Read the following chapter of *An introduction to statistical learning*:

- 3.5 Comparison of Linear Regression with K-Nearest Neighbors

2.3 Practical session

Task - Fit a knn regression

With the bmd.csv dataset, we want to fit a knn regression with k=3 for BMD, with age as covariates. Then we will compute the MSE and R^2.

```
#libraries that we will need
library(FNN)   #knn regression
```

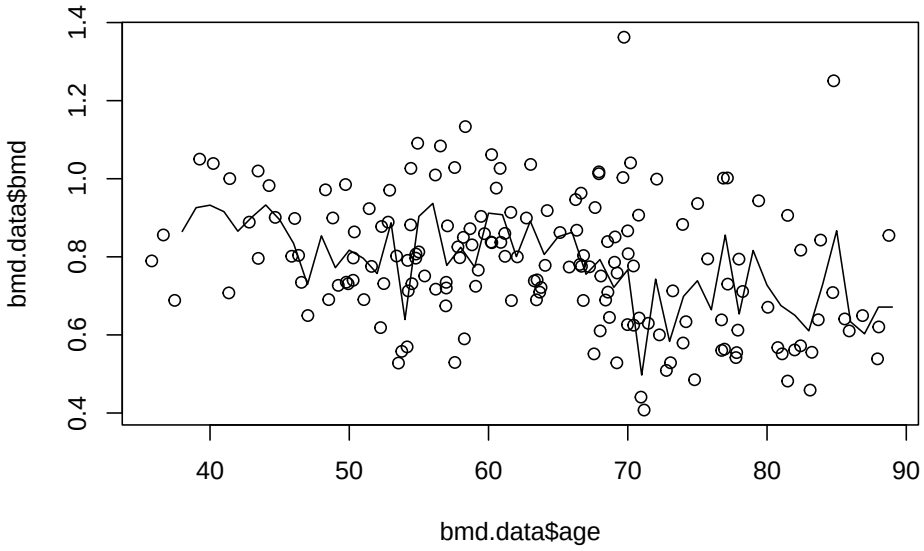
```
## Warning: package 'FNN' was built under R version 4.3.3
```

```
set.seed(1974) #fix the random generator seed

#read the data
bmd.data     <-
  read.csv("https://www.dropbox.com/s/c6mhgatkotuze8o/bmd.csv?dl=1",
           stringsAsFactors = TRUE)
```

```
#Fit a knn regression with k=3
#using the knn.reg() function from the FNN package
knn3.bmd <- knn.reg(train=bmd.data[c("age")],
                    y=bmd.data$bmd,
                    test= data.frame(age=seq(38,89)),
                    k=3)
```

Before computing the MSE and R^2, we will plot the model predictions

```
  plot(bmd.data$age, bmd.data$bmd) #adding the scatter for BMI and BMD
  lines(seq(38,89), knn3.bmd$pred)  #adds the knn k=3 line
```



Finally, we compute the MSE and R^2 for knn k=3. We have to refit the models and "test" them in the original data

```
  knn3.bmd.datapred <- knn.reg(train=bmd.data[c("age")],
                               y=bmd.data$bmd,
                               test=bmd.data[c("age")], #ORIGINAL DATA
                               k=3)       #knn reg with k=3
  #MSE for knn k=3
  mse.knn3  <- mean((knn3.bmd.datapred$pred - bmd.data$bmd)^2)
```

```
  mse.knn3
```

```
## [1] 0.01531282
```

```
  r2.knn3   <- 1- mse.knn3/(var(bmd.data$bmd)*168/169) #R2 for knn k=3 using
  r2.knn3                                              #R2 = 1-MSE/var(y)
```

```
## [1] 0.4445392
```

TRY IT YOURSELF:

1) Fit a knn regression for BMD using AGE, with k=20

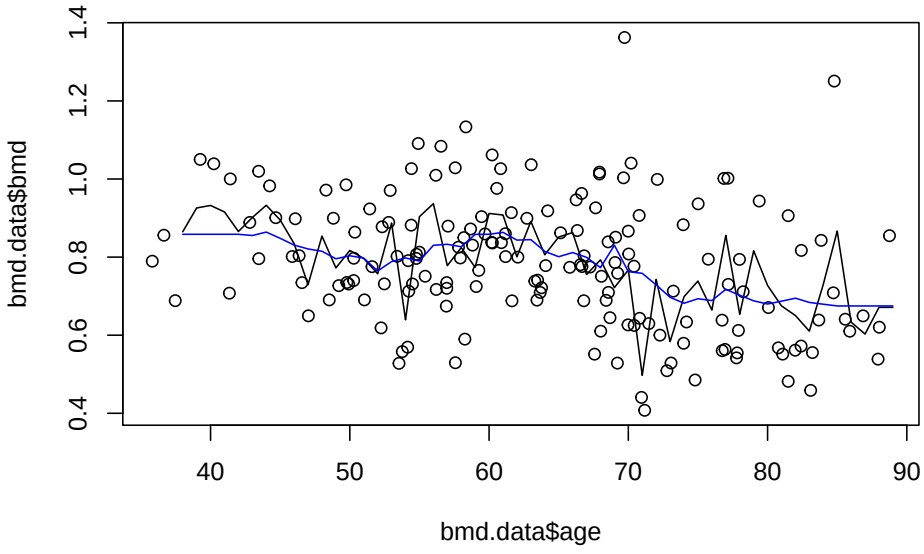
See the solution code

```
  knn20.bmd <- knn.reg(train=bmd.data[c("age")],
                       y=bmd.data$bmd,
                       test= data.frame(age=seq(38,89)),
                       k=20)     #knn regression with k=20
```

2) Add the prediction line to the previous scatter plot

See the solution code

```
plot(bmd.data$age, bmd.data$bmd) #adding the scatter for BMI and BMD
lines(seq(38,89), knn3.bmd$pred)
lines(seq(38,89), knn20.bmd$pred, col="blue")  #adds the knn k=20 gray line
```



3) Compute the MSE and R^2

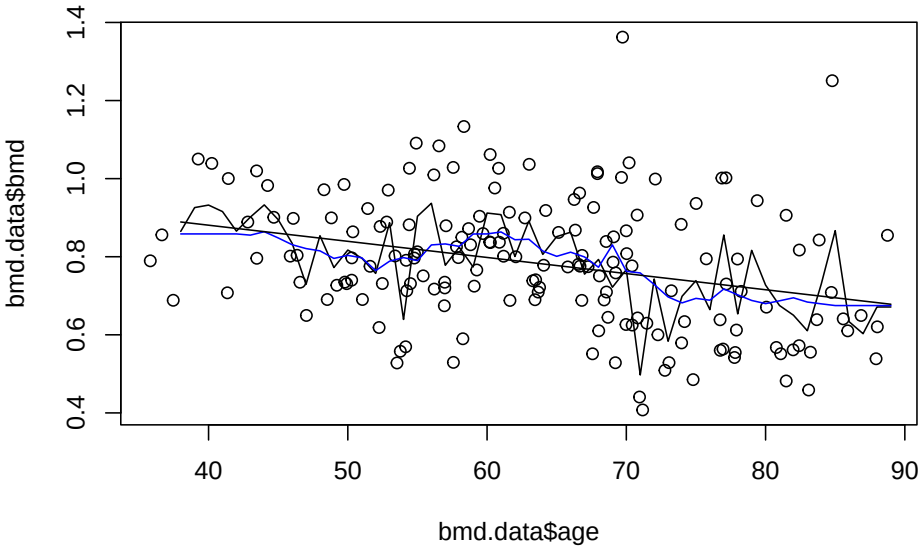
See the solution code

```r
#predictions from knn reg with k=20
knn20.bmd.datapred <- knn.reg(
  train = bmd.data[c("age")],
  y = bmd.data$bmd,
  test = bmd.data[c("age")],
  #ORIGINAL DATA
  k = 20
)

#MSE for knn k=20
mse.knn20 <- mean((knn20.bmd.datapred$pred - bmd.data$bmd) ^ 2)
mse.knn20

#r2 for knn k=20
r2.knn20 <- 1-mse.knn20/(var(bmd.data$bmd)*168/169) #168/169 is added to correct
r2.knn20                                             #the degrees of freedom
```

4) Add a linear prediction for BMD using AGE to the scatter plot

See the solution code

```r
#The initial plot with knn3 and knn20
plot(bmd.data$age, bmd.data$bmd) #adding the scatter for BMI and BMD
lines(seq(38,89), knn3.bmd$pred)
lines(seq(38,89), knn20.bmd$pred, col="blue")  #adds the knn k=20 gray line

#linear model and predictions
model.age <- lm(bmd ~ age,
                data = bmd.data)
bmd.pred <- predict(model.age,
                newdata = data.frame(age=seq(38,89)))
lines(seq(38,89), bmd.pred) #add the linear predictions to the plot
```



2.4 Exercises

1) With the *fat* dataset in the *library(faraway)*, we want to predict body fat (variable **brozek**) using the variable **abdom**

   a) use a k-nearest neighbour regression, with k=3,5 and 11, to approximate the relation between **brozek** and **abdom**. Plot the three lines.

   b) What is the predicted **brozek** for someone with **abdom=90** using knn=11?

   c) What is the predicted **brozek** for someone with **abdom=90** using a linear model?

   d) Compute the mean squared error for k=3,5 and 11

2) With the same data *fat*, use a knn (k=9) to predict body fat (variable **brozek**) using the variables **abdom** and **age**

a) Plot the predictions for **abdom = 80 to 115** and **age = 30**

b) Would you expect the mean squared error for k=1 to be greater or smaller than the one for k=9? Why would you prefer k=9 over k=1?

Chapter 3

Logistic regression

3.1 Introduction

You should also be familiar with logistic regression but not necessarily as a classification method. In this section, we will see how this model can be used to make predictions for categorical outcomes. Like the linear model, there will be several aspects, such as hypothesis testing, that we will not discussed in detail.

3.2 Readings

Read the following chapters of *An introduction to statistical learning*:

- 4.2 Why not Linear Regression?
- 4.3 Logistic Regression
- Read about the *Confusion Matrix* and *ROC curve* in the subchapter 4.4.2

3.3 Practical session

Task - Logistic regression

With the bmd.csv dataset, let's fit a logistic regression model to predict fracture, using AGE, SEX, BMI and BMD as main effects.

```
#libraries that we will need
library(pROC) #ROC curve
set.seed(1974) #fix the random generator seed

#read the data
bmd.data     <-
  read.csv("https://www.dropbox.com/s/c6mhgatkotuze8o/bmd.csv?dl=1",
```

```
           stringsAsFactors = TRUE)

bmd.data$bmi <- bmd.data$weight_kg / (bmd.data$height_cm/100)^2
```

```
#Fits a logistic model with fixed effects only
  model1.fracture <- glm(fracture=="fracture" ~ age + sex + bmi + bmd,
                         family=binomial, data = bmd.data)
  summary(model1.fracture)
```

```
##
## Call:
## glm(formula = fracture == "fracture" ~ age + sex + bmi + bmd,
##     family = binomial, data = bmd.data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   9.79488    2.69720   3.631 0.000282 ***
## age           0.01844    0.02094   0.881 0.378540
## sexM          0.84599    0.51249   1.651 0.098792 .
## bmi          -0.05131    0.06013  -0.853 0.393537
## bmd         -15.11747    2.80337  -5.393 6.94e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 205.27  on 168  degrees of freedom
## Residual deviance: 110.86  on 164  degrees of freedom
## AIC: 120.86
##
## Number of Fisher Scoring iterations: 6
```

We will use a Bayes classifier threshold (pred prob <.5) to classify each patient and then check the misclassifications using the confusion matrix.

```
#for model1
model1.fracture.pred <- predict(model1.fracture,      #probabilities
                                type="response")      #predicted by the model

#if prob>.5 returns TRUE (fracture)
model1.fracture.class <- model1.fracture.pred > .5

#now build the confusion matrix
table(model1.fracture.class, bmd.data$fracture)
```

```
##
## model1.fracture.class fracture no fracture
```

```
##                  FALSE        15          110
##                  TRUE         35            9
```

So, from the table above you can see that based on the cut-off for the predictive probability of 0.5, the model predicted 35 out of the 50 fractures. And it predicted 110 out of the 119 no fractures.

Let's now plot the ROC and calculate the area under the curve. There are several packages in R to do this; we will be using the library pROC.

```
  auc.model1.fracture <- roc(fracture ~ model1.fracture.pred, #using the pred prob
                             data = bmd.data)                 #from the model
```

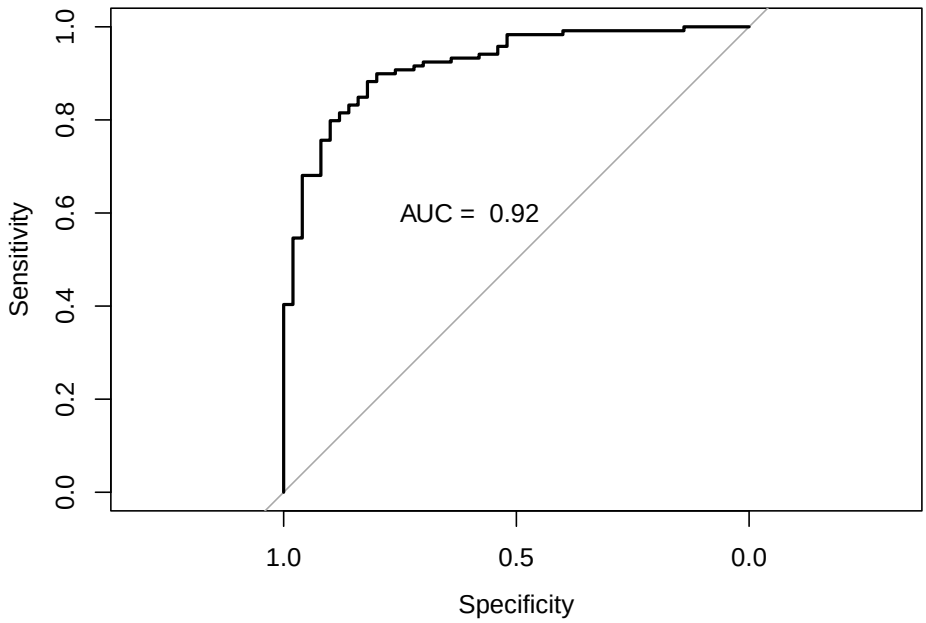
```
## Setting levels: control = fracture, case = no fracture
```

```
## Setting direction: controls > cases
```

```
  auc.model1.fracture
```

```
##
## Call:
## roc.formula(formula = fracture ~ model1.fracture.pred, data = bmd.data)
##
## Data: model1.fracture.pred in 50 controls (fracture fracture) > 119 cases (fracture no fractur
## Area under the curve: 0.9195
```

```
  plot(auc.model1.fracture)
  text(0.6,.6, paste("AUC = ", round(auc.model1.fracture$auc,2)))
```



TRY IT YOURSELF: 1) Fit a similar model to model1.fracture but add a

quadratic effect for age, i.e., I(age^2) and compare the AIC of both models.

See the solution code

```
#Fits a logistic model as model1.fracture but adds the quadratic effect for age
    model2.fracture <- glm(fracture=="fracture" ~ age +  I(age^2) + bmi +
                                                  sex + bmd,
                           family=binomial, data = bmd.data)
    summary(model2.fracture)
```

2) Compute the classification error for the Bayes classifier using the confusion matrix.

See the solution code

```
#Produce the confusion matrices for the models fitted
model2.fracture.pred <- predict(model2.fracture,     #probabilities predicted
                                type="response")     #by the model

#if prob>.5 returns TRUE (fracture)
model2.fracture.class <- model2.fracture.pred >.5

#confusion matrix
table(model2.fracture.class, bmd.data$fracture)
```

3) Plot the ROC curve

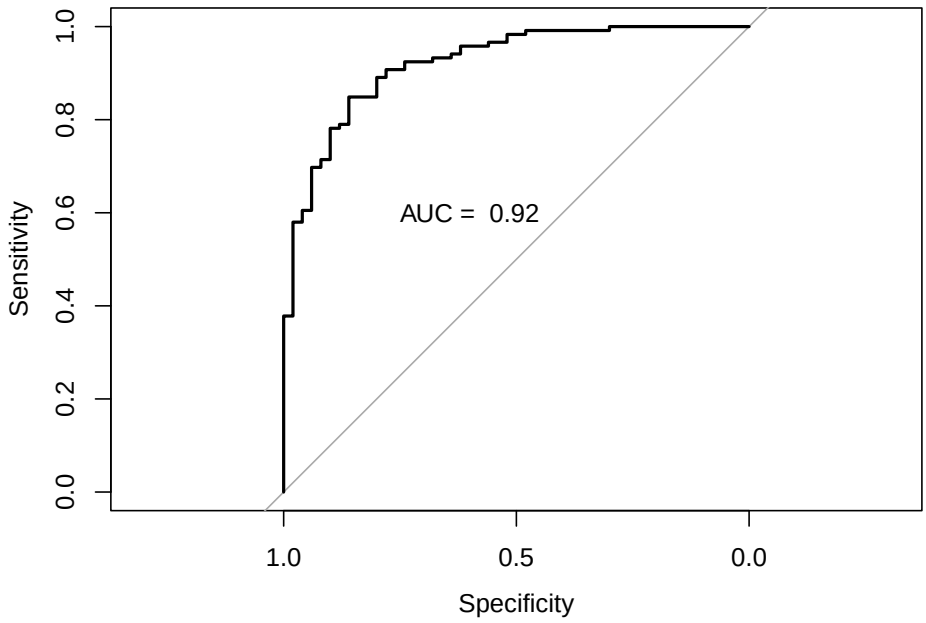
See the solution code

```
#ROC
  #using the pred prob from the model
  auc.model2.fracture <- roc(fracture ~ model2.fracture.pred,
                             data = bmd.data)
```

```
## Setting levels: control = fracture, case = no fracture
```

```
## Setting direction: controls > cases
```

```
  auc.model2.fracture
  plot(auc.model2.fracture)
  text(0.6,.6, paste("AUC = ", round(auc.model2.fracture$auc,2)))
```



3.4 Exercises

1) The dataset bdiag.csv, included several imaging details from patients that had a biopsy to test for breast cancer.
The variable **Diagnosis** classifies the biopsied tissue as M = malignant or B = benign.

   a) Fit a logistic regression to predict **Diagnosis** using **texture_mean** and **radius_mean**.

   b) Build the confusion matrix for the model above

   c) Calculate the area and the ROC curve for the model in a).

   d) Plot the scatter plot for **texture_mean** and **radius_mean** and draw the border line for the prediction of **Diagnosis** based on the model in a)

   e) If you wanted to use the model above to predict the result of the biopsy, but wanted to decrease the chances of a false negative test, what strategy could you use?

2) The SBI.csv dataset contains the information of more than 2300 children that attended the emergency services with fever and were tested for serious bacterial infection. The outcome **sbi** has 4 categories: Not Applicable(no infection) / UTI / Pneum / Bact

a) Build a multinomial model using **wcc**, **age**, **prevAB**, **pct**, and **crp** to predict **sbi**

b) Compute the confusion matrix and compute the kappa statistics

c) How does the model classify a child with 1 year of age, WCC=29, PCT=5, CRP=200 and no prevAB?

Chapter 4

Linear Discriminant Analysis

4.1 Introduction

Once again we focus on $\Pr(Y = k|X = x)$ to classify an individual (or other unit) in one of the categories of Y. Using the Bayes theorem,

$\Pr(Y = k|X = x) = \frac{f_k(x)\Pr(Y=k)}{f(x)}$,

where $f_k(x)$ is the density for $X \mid Y = k$.

Thus, finding the category k that has the highest probability $\Pr(Y = k|X = x)$ is the same as finding the category k with higher value for $\frac{f_k(x)\Pr(Y=k)}{f(x)}$.

Now, if we assume that the density of X (represented above in a slight abuse of notation as Pr(X=x)) is $N(\mu, \sigma^2)$, it is possible to show that maximising the right side of the equation is equivalent to maximising

$$\underbrace{x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log\big(\Pr(Y = k)\big)}_{\text{discriminant function}}$$

So, if we get estimates for the parameters in the discriminant function, we can calculate the category k that has the highest discriminant value and thus the highest $\Pr(Y = k|X = x)$.

4.2 Readings

Read the following chapters of *An introduction to statistical learning*:

- 4.4 Linear Discriminant Analysis

- 4.5 A Comparison of Classification Methods

4.3 Practical session

Task 1 - Classification with the linear discriminant function

With the bmd.csv dataset, let's use the variable **bmd** to predict **fracture** using linear discriminant analysis.

The discriminant function is given by

$age \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log\big(\Pr(Y = k)\big)$,

where μ_k is the mean **bmd** for the group $k =$ "fracture" or $k =$ "no fracture", σ is the standard deviation for **bmd**, and $\Pr(Y = k)$ is the (marginal) probability of each category of the outcome.

We can easily get estimates for these parameters.

```
#libraries that we will need
library(MASS) #lda function
set.seed(1974) #fix the random generator seed

#read the data
bmd.data     <-
  read.csv("https://www.dropbox.com/s/c6mhgatkotuze8o/bmd.csv?dl=1",
           stringsAsFactors = TRUE)
```

```
#mean bmd for fracture
mean.f   <- with(bmd.data, mean(bmd[fracture=="fracture"]))
#mean bmd for no fracture
mean.nf  <- with(bmd.data, mean(bmd[fracture=="no fracture"]))
#estimate of sigma (see page 141)
sigma.bmd <- sqrt(with(bmd.data,
                       (sum((bmd[fracture=="fracture"] - mean.f)^2) +
                       sum((bmd[fracture=="no fracture"]- mean.nf)^2))/
                    (length(bmd)-2)
                    )
                  )
#probability of fracture/no fracture
pr.fracture <- prop.table(table(bmd.data$fracture))

print(c(mean.f, mean.nf, sigma.bmd))
```

```
## [1] 0.6233080 0.8502454 0.1305394
```

```
print(pr.fracture)
```

```
##
```

```
##    fracture no fracture
##    0.295858    0.704142
```

So, now we can compute the value of the discriminant function for a particular **bmd**. For example, for **bmd**=0.54

```
#for fracture
0.54*mean.f/sigma.bmd^2 - mean.f^2/(2*sigma.bmd^2) + log(pr.fracture[1])
```

```
## fracture
## 7.134559
```

```
#for no fracture
0.54*mean.nf/sigma.bmd^2 - mean.nf^2/(2*sigma.bmd^2) + log(pr.fracture[2])
```

```
## no fracture
##    5.381084
```

Thus, for **bmd**=0.54, the classification would "fracture" given that this category has the highest value for the discriminant function.

The linear discriminat analysis is implemented in the `lda()` function from the `library(MASS)`

```
library(MASS)
lda.model <- lda(fracture~bmd, data=bmd.data)

#to classify someone with bmd=-.54
#
predict(lda.model, newdata=data.frame(bmd=0.54))$class
```

```
## [1] fracture
## Levels: fracture no fracture
```

TRY IT YOURSELF:

1) Compute the confusion matrix for LDA using **bmd** to predict **fracture**.

See the solution code

```
 #predictions
pred.dataset <- predict(lda.model)$class
#confusion matrix
table(bmd.data$fracture, pred.dataset)
```

2) Additionally to **bmd**, use **age**, **weight_kg** and **height_cm**, to predict **fracture** using LDA, and compute the confusion matrix. Compare the kappa statistic for this result with the one obtained in 1).

See the solution code

```
lda.model2 <- lda(fracture~bmd+age+weight_kg+height_cm,
                  data=bmd.data)
```

```
#predictions
pred.dataset2 <- predict(lda.model2)$class
#confusion matrix
table(bmd.data$fracture, pred.dataset2)

library(irr) #for the kappa statistics
```

```
## Loading required package: lpSolve
```

```
kappa2(cbind(bmd.data$fracture, pred.dataset))    #model in 1)
kappa2(cbind(bmd.data$fracture, pred.dataset2))   #current model
```

Task 2 - Classification with the quadratic discriminant function

The linear discriminant function assumes that the variance is the same for all the categories of the outcome. The quadratic discriminant analysis (QDA) relaxes this assumption.

Let's repeat the classification of **fracture** with **bmd**, using a QDA

```
#qda() is a function from the MASS
#library that fits QDA
qda.model <- qda(fracture ~ bmd,
                 data=bmd.data)
```

We can now predict **fracture** for the individuals in the dataset and compare it with the observed values (confusion matrix)

```
#predictions
pred.qda <- predict(qda.model)$class
#confusion matrix
table(bmd.data$fracture, pred.qda)
```

```
##              pred.qda
##               fracture no fracture
##   fracture          34          16
##   no fracture       10         109
```

TRY IT YOURSELF:

1) Additionally to **bmd**, use **age**, **weight_kg** and **height_cm**, to predict **fracture** using QDA, and compute the confusion matrix.

See the solution code

```
qda.model2 <- qda(fracture~bmd+age+weight_kg+height_cm,
                  data=bmd.data)
#predictions
pred.qda2 <- predict(qda.model2)$class
```

```r
#confusion matrix
table(bmd.data$fracture, pred.qda2)
```

4.4 Exercises

1) The dataset bdiag.csv, included several imaging details from patients that had a biopsy to test for breast cancer.
The variable **Diagnosis** classifies the biopsied tissue as M = malignant or B = benign.

   a) Use LDA to predict **Diagnosis** using **texture_mean** and **radius_mean**.

   b) Build the confusion matrix for the model above

   c) Compare the results with a logistic regession

   d) Plot the scatter plot for **texture_mean** and **radius_mean** and draw the border line for the prediction of **Diagnosis** based on the model in a)

   e) Use **radius_mean, texture_mean, perimeter_mean, area_mean, smoothness_mean, compactness_mean, symmetry_mean, fractal_dimension_mean** to classify **diagnosis** with LDA and QDA. Check the distribution of the predictors.

2) Exercise 5 from the book (page 191)

Chapter 5

K-nearest Neighbours Classification

5.1 Introduction

The K-nearest Neighbours (KNN) for classification, uses a similar idea to the KNN regression. For KNN, a unit will be classified as the majority of its neighbours.

5.2 Readings

Read the following chapters of *An introduction to statistical learning*:

- 4.1 An Overview of Classification
- 2.2.3 The Classification Setting

5.3 Practical session

Task - KNN classification

With the bmd.csv dataset, we will use KNN (k=3) with the variables AGE, SEX, BMI and BMD to classify FRACTURE and compute the confusion matrix

First, let's import the dataset

```
#libraries that we will need
library(class)   #knn
set.seed(1974) #fix the random generator seed

#read the data
```

```
bmd.data     <-
  read.csv("https://www.dropbox.com/s/c6mhgatkotuze8o/bmd.csv?dl=1",
           stringsAsFactors = TRUE)

bmd.data$bmi <- bmd.data$weight_kg / (bmd.data$height_cm/100)^2
```

let's standardise all the variables so they have mean 0 and SD=1 so that the distances are in the same scale.

```
  bmd.data$age.std     <- scale(bmd.data$age)
  bmd.data$sex.num.std <- scale(as.numeric(bmd.data$sex)) #1-F, #2-M
  bmd.data$bmi.std     <- scale(bmd.data$bmi)
  bmd.data$bmd.std     <- scale(bmd.data$bmd)
```

Now we use knn=3

```
model.knn3 <-  knn(train = bmd.data[c("age.std", "sex.num.std",
                                      "bmi.std","bmd.std")],
                   test  = bmd.data[c("age.std", "sex.num.std",
                                      "bmi.std","bmd.std")],
                    cl    = bmd.data$fracture, k=3)
table(model.knn3, bmd.data$fracture)
```

```
##
## model.knn3    fracture no fracture
##   fracture          38           7
##   no fracture       12         112
```

TRY IT YOURSELF:

1) Repeat the classification model from above but now with k=20 and compute the confusion matrix.

See the solution code

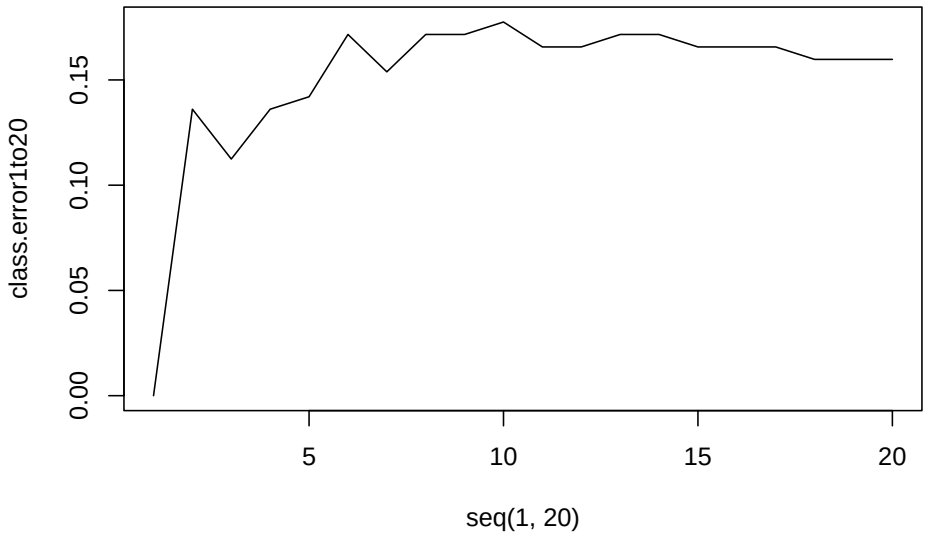
```
  model.knn20 <-  knn(train = bmd.data[c("age.std", "sex.num.std",
                                         "bmi.std","bmd.std")],
                      test  = bmd.data[c("age.std", "sex.num.std",
                                         "bmi.std","bmd.std")],
                       cl    = bmd.data$fracture, k=20 )
  table(model.knn20, bmd.data$fracture)
```

2) (advanced) Plot the error rate for KNN with k=1 to 20 using the variables AGE, SEX, BMI and BMD to classify FRACTURE

See the solution code

```
knn.fit <- function(k.par){
  knn.model <- knn(train = bmd.data[c("age.std", "sex.num.std",
                                      "bmi.std","bmd.std")],
```

```
                    test  = bmd.data[c("age.std", "sex.num.std",
                                       "bmi.std","bmd.std")],
                     cl    = bmd.data$fracture, k=k.par )
  class.error<- 1-sum(diag(table( knn.model, bmd.data$fracture))/169)
  return(class.error)
}

  class.error1to20 <- sapply(seq(1,20), knn.fit)
  plot(seq(1,20), class.error1to20, type="l")
```



5.4 Exercises

1) The dataset bdiag.csv, included several imaging details from patients that had a biopsy to test for breast cancer.
   The variable **diagnosis** classifies the biopsied tissue as M = malignant or B = benign.

   a) Use a KNN with k=5 to predict **Diagnosis** using **texture_mean** and **radius_mean**.

   b) Build the confusion matrix for the classification above

   c) Plot the scatter plot for **texture_mean** and **radius_mean** and draw the border line for the prediction of **Diagnosis** based on the model in a)

   d) Plot the scatter plot for **texture_mean** and **radius_mean** and draw the border line for the prediction of **Diagnosis** based knn, k=15