Compulsory exercise 1: Group 3

TMA4268 Statistical Learning V2018

Joakim Olsen, Mathias Opland and Lars Fredrik Espeland

16.02.2018

Problem 1 - Core concepts in statistical learning

a) Training and test MSE

- We see that for K=1, the variance is pretty high, which is expected as the curves only will go from point to point. At K=2 we see that the variance lowers some as we now take into account the two nearest points, and the peaks will not be that far away from the true line. When K increases, we take account for neigbhours which is not necessary very local anymore. Which K to choose to keep the neighbourhood local depends on the number of observations in the training set. Here n=61, and already when K=10, we get somewhat of a problem in the end points. Although we get a very good fit along most of the line, at the end it will take the same points into account, and one will get a straight line. At K=61, one will only get a straight line, which is the mean.
- A small value of K will give a more flexible fit, which has low bias but high variance. As mentioned above, K=1 will give point-to-point plots, which is an overfit. At K=2 we get a lower variance, and there is no sign of underfitting the curve. At K=10 the variance is notably lower, but the plot shows signs of underfitting and a low flexibility. Continuing the trend, at K=25 we get a clearly underfitted and inflexible result.
- The MSE for the training sets show how close the predicted curve is to the points it is based on. As we can see, at K=1 the MSE is zero, which is a clear sign of overfitting. The MSE just increases as K increases, which indicates low flexibility. There is by this plot hard to say where the overfitting stops and the low flexibility starts, and thus hard to conclude what is the best value for K. On the other hand, the MSE for the test set decreases for the first K's, and then increases. The test set is not affected by overfitting in the same way, and will therefore show the best value for K much clearer.
- The lowest point, being around K=5 here, is where the error is lowest and would be our choice for K.

b) Bias-variance trade-off

- We have a true underlying curve expressed as $Y = f(x) + \epsilon$. Here ϵ is the random or irreducible error, and we assume it has mean zero and constant variance equal to σ^2. The expected test mean squared error (MSE) for x_0 is defined as:

  $$E[Y - \hat{f}(x_0)]^2.$$

  Then we decompose the MSE into three terms:

  $$E[(Y - \hat{f}(x_0))^2] = Var(\epsilon) + Var[\hat{f}(x_0)] + [Bias(\hat{f}(x_0))]^2.$$

  $Var[\hat{f}(x_0)]$ can be found by taking a sample variance over the M repeated training sets. This variance will be based on the sample mean from the data sets, and is therefore unrelated to the true underlying curve. The bias is however calculated as the difference between the sample mean and the true underlying curve, and this picks up the variance to the true underlying curve.
- Interpretation of Figure 4:

  - The squared bias decreases as the flexibility increases. That means that for high flexibility there is no systematic difference from the predicted model and the values being estimated. At higher values for K there will be a systematic difference. As example, in a) when K=61, we will always get a straight line, which differs from the true values.
  - The variance decreases as K increases, since more points are taken account for, and the difference from each training set to the others is less.
  - The irreducible error is not dependent on the values of K, and is therefore constant.

- Based on the sum of the errors, the lowest error is for K=3 (but nearly the same as K=5). Thus this will be our choice for K as this will give the most accurate model. This agrees fairly well with what we found in a), and our conclusion would be that a K between 3 and 5 would give the best results.

Problem 2 - Linear regression

a) Understanding model output

- In this problem, we are going to model $-\frac{1}{\sqrt{SYSBP}}$ as a function of the covariates SEX, AGE, CURSMOKE, BMI, TOTCHOL and BPMEDS, using the data from n = 2600 persons. The equation for the fitted `modelA`, where $-\frac{1}{\sqrt{SYSBP}}$ is the response, is

$$\boldsymbol{Y} = \boldsymbol{\beta X} + \boldsymbol{\epsilon},$$

where $\boldsymbol{\beta}$ is a vector consisting of the regression parameters corresponding to each covariate, including the intercept, β_0, and $\boldsymbol{\epsilon}$ is the error term, assumed to be normally distributed with mean 0.

```
modelA=lm(-1/sqrt(SYSBP) ~ .,data = data)
summary(modelA)
```

```
##
## Call:
## lm(formula = -1/sqrt(SYSBP) ~ ., data = data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0207366 -0.0039157 -0.0000304  0.0038293  0.0189747
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.103e-01  1.383e-03 -79.745  < 2e-16 ***
## SEX         -2.989e-04  2.390e-04  -1.251 0.211176
## AGE          2.378e-04  1.434e-05  16.586  < 2e-16 ***
## CURSMOKE    -2.504e-04  2.527e-04  -0.991 0.321723
## BMI          3.087e-04  2.955e-05  10.447  < 2e-16 ***
## TOTCHOL      9.288e-06  2.602e-06   3.569 0.000365 ***
## BPMEDS       5.469e-03  3.265e-04  16.748  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005819 on 2593 degrees of freedom
## Multiple R-squared:  0.2494, Adjusted R-squared:  0.2476
## F-statistic: 143.6 on 6 and 2593 DF,  p-value: < 2.2e-16
```

- In the summary output, the column with `Estimate` is the estimated values for the coefficients of the model, the $\hat{\beta}'s$. In particular, `Intercept` is the estimate for β_0. This is the value of the response when

all the covariates are 0. In this case, this value is not that meaningful, since it for instance makes no sense to have a person with a BMI of 0. The other estimated coefficients tell us how much we can expect $-\frac{1}{\sqrt{SYSBP}}$ to change given a change in the respective covariate when all the other covariates are fixed. The parameters in $\boldsymbol{\beta}$ can be estimated with maximum likelihood and least squares, which both gives

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}.$$

- `Std. Error` is the standard error of the coefficients. These values tell us how close the estimated coefficients are to the true values of the coefficients. The standard error of each estimated coefficient is found by taking the square root of the corresponding term on the diagonal of the covariance matrix $\Sigma = \sigma^2(\boldsymbol{X}^T\boldsymbol{X})$, where σ^2 is the variance of the resiudals.
- Further, the values under `t value` tell us how many standard deviations β_i is away from 0. For each covariate, we investigate whether there is a relationship between the covariate and the response. This is done through the hypothesis test $H_0 : \beta_i = 0$ versus $H_1 : \beta_i \neq 0$. Then the t-statistic is calculated by

  $$t_i = \frac{\hat{\beta}_i - 0}{SE(\hat{\beta}_i)}.$$

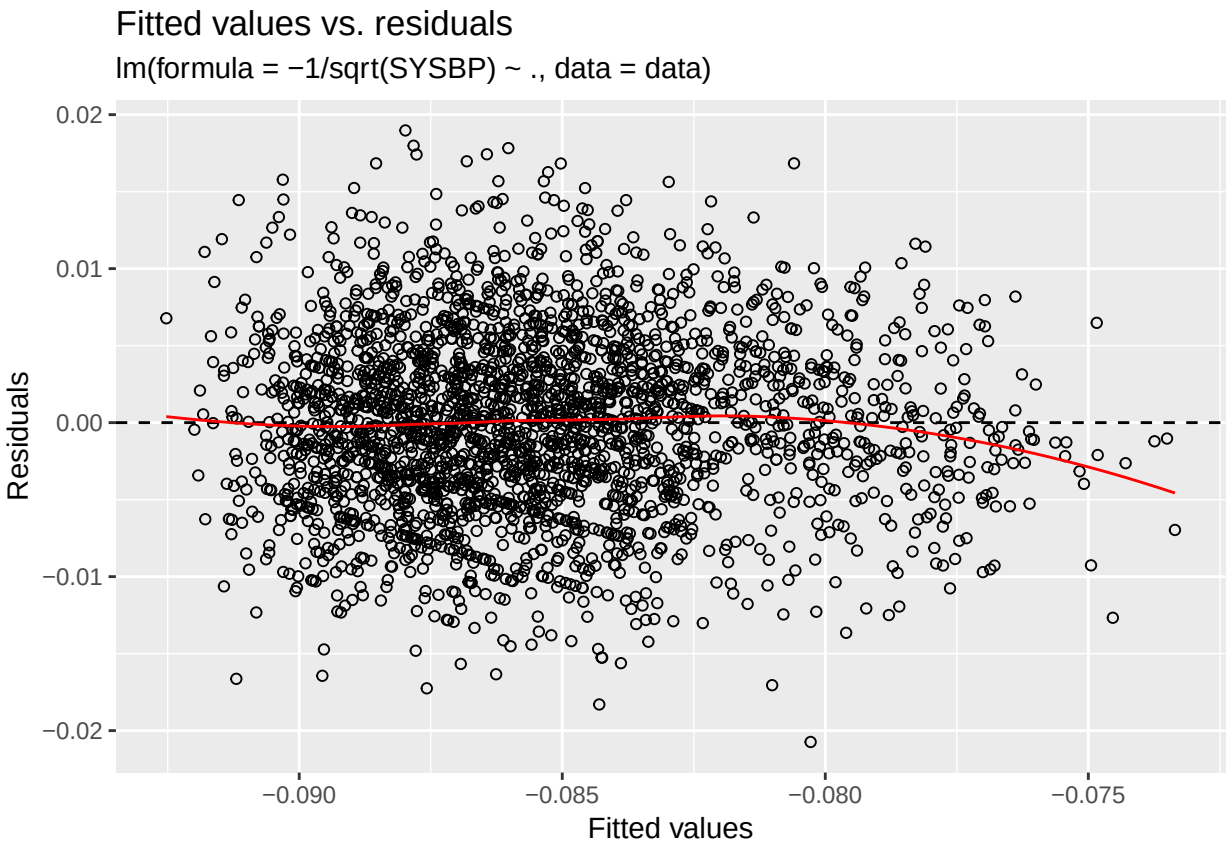
  This is the formula for the t-value of each covariate. We can see that if the t-value is large, the variable is most likely related to the response.
- If there is no relation between the variable and the response, then the t-statistic is expected to have a t-distribution with n-2 degrees of freedom. Due to the Central Limit Theorem, the t-distribution will be quite similar to the normal distribution for values of n larger than approximately 30. Thus, it is easy to compute the probability of observing any number larger than $|t|$. These probabilities are the p-values, and they are shown in the column under `Pr(>|t|)` in the summary output. Small p-values tell us that it is very unlikely that the observed association between the variable and the response is only due to chance. Thus the p-values indicate whether each of the variables is related to the response, so we can find out whether the hypothesis test described above is significant or not.
- The `Residual standard error` is an estimate of the standard deviation of the error term ϵ. We have that $\sigma^2 = Var(\epsilon)$, so the Residual standard error, RSE, is an estimate of σ. The formula for this is $RSE = \sqrt{RSS/(n-2)}$, where RSS is the residual sum of squares, $RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)$. The residual standard error is an absolute measurement of how much the model deviates from the true data.
- In order to see whether the regression is significant, that means whether there is a relationship between the response and the covariates at all, one can use another hypothesis test. If the null hypothesis is set to be $H_0 : \beta_{SEX} = \beta_{AGE} = ... = \beta_{BPMEDS} = 0$, with the alternative H_1 : at least one β_j is non-zero, the `F-statistic` can be computed. The formula for this is

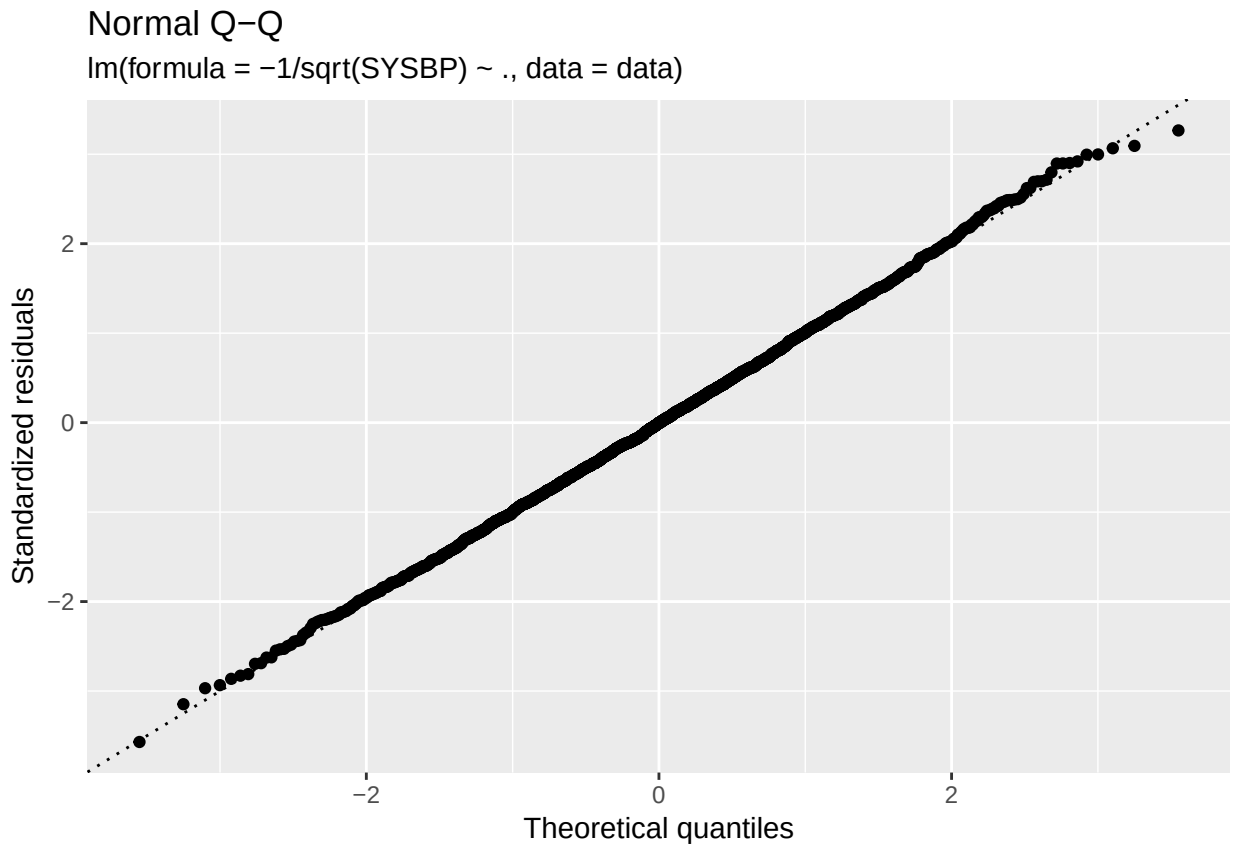
  $$F = \frac{(TSS - RSS)/p}{RSS/(n-p-1)},$$

  where $TSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)$. If the F-statistic is close to 1, it may be an indication of an association between the response and the predictors. Here, the F-statistic is much larger than 1, so we can expect the alternative hypothesis to be true.

b) Model fit

- While the residual standard error gives an absolute measure of lack of fit of the model, the R^2-statistic provides a value between 0 and 1. To calculate this, we use the formula $R^2 = 1 - \frac{RSS}{TSS}$. This value is the proportion of variance which the model can explain. We can say that TSS is a measurement of the total variance in the response Y, while RSS is the amount of variability that the regression model does not explain. Thus, the R^2-value becomes the proportion of variance which the model is able to explain. Here, the R^2-value of the data is 0.2494. In other words, about 1/4 of the variability can be explained by the fitted `modelA`.


Fitted values vs. residuals
lm(formula = −1/sqrt(SYSBP) ~ ., data = data)
0.02
0.01
0.00
−0.01
−0.02
Residuals
−0.090
−0.085
−0.080
−0.075
Fitted values



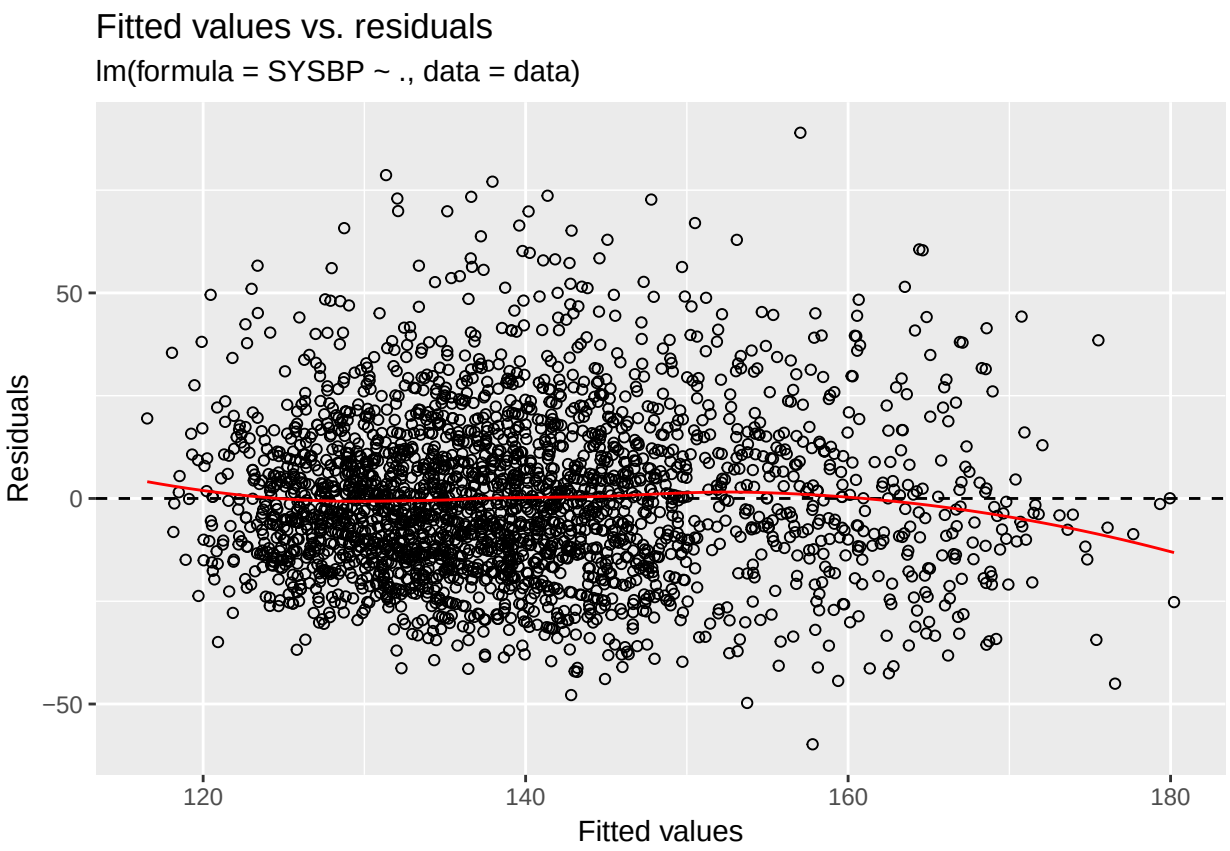
- Diagnostic plots of fitted values against standardized residuals and QQ-plot of standardized residuals for modelA are shown above. From these plots, we can see that the model seems to fit well. The plot of the residuals vs fitted values shows that the residuals, except for the largest fitted values, are equally spread around a horizontal line. This is an indication of a linear relationship between the response and the covariates. In addition, the residuals in the qq-plot follow almost a straight line all the way. Thus, the residuals seem to be Gaussian distributed.
- We then make a new model, modelB, where SYSBP is the response:

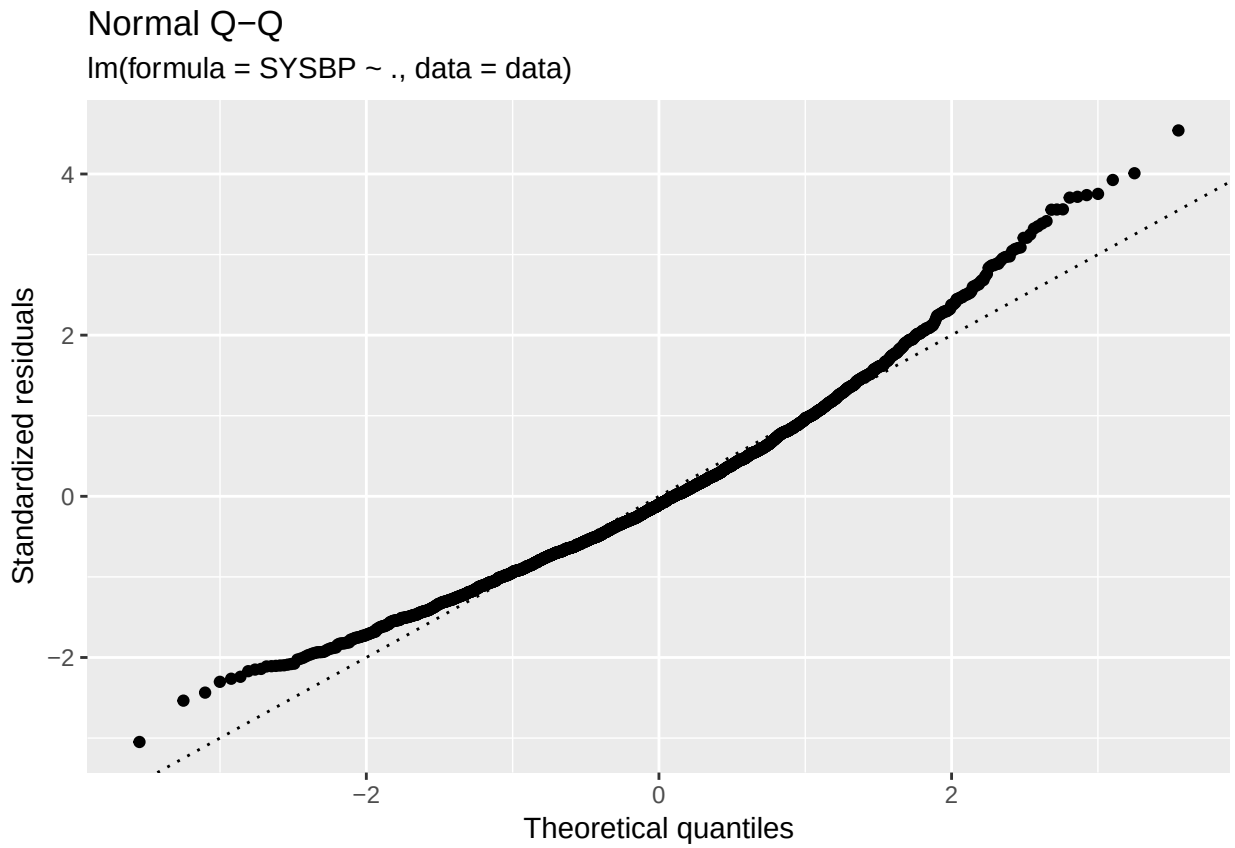
```
modelB=lm(SYSBP ~ .,data = data)
summary(modelB)
```

```
##
## Call:
## lm(formula = SYSBP ~ ., data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -59.800 -13.471  -1.982  11.063  88.959
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 56.505170   4.668798  12.103  < 2e-16 ***
## SEX         -0.429973   0.807048  -0.533  0.59424
## AGE          0.795810   0.048413  16.438  < 2e-16 ***
## CURSMOKE    -0.518742   0.853190  -0.608  0.54324
## BMI          1.010550   0.099770  10.129  < 2e-16 ***
```

```
## TOTCHOL      0.028786   0.008787   3.276  0.00107 **
## BPMEDS      19.203706   1.102547  17.418  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.65 on 2593 degrees of freedom
## Multiple R-squared:  0.2508, Adjusted R-squared:  0.249
## F-statistic: 144.6 on 6 and 2593 DF,  p-value: < 2.2e-16
```

For this model, the plot of residuals against fitted values and the qq-plot are as follows:





The plot of fitted values vs residuals for modelB seems quite similar to the one for modelA. However, although the residuals mostly are spread around a horizontal line, it is seemingly not the case for all the largest and smalles fitted values. It is however difficult to separate the two models based on this plot. Therefore, the qq-plot gives more information. We can see that the residuals do not quite follow a straight line, so the residuals are probably not Gaussian distributed.

In addition, we can do the Anderson-Darling normality test for the two models:

```
library(nortest)
ad.test(rstudent(modelA))
```

```
##
##  Anderson-Darling normality test
##
## data:  rstudent(modelA)
## A = 0.19209, p-value = 0.8959
```

```
ad.test(rstudent(modelB))
```

```
##
##  Anderson-Darling normality test
##
## data:  rstudent(modelB)
## A = 13.2, p-value < 2.2e-16
```

This test tells us how well the data follow the Gaussian distribution. If the distribution does not fit the data, the p-value will be small. From the p-values here, we can clearly conclude that the residuals in modelB are not Gaussian distributed, while the opposite is most likely true in modelA.

Due to these comparisons, we would prefer modelA if the aim is to make inference about systolic blood pressure. Unlike the residuals in modelB, the transformation of the response to $-\frac{1}{\sqrt{SYSBP}}$ gives residuals which are almost Gaussian distributed, so this one is preferable.

c) Confidence interval and hypothesis test

- Using modelA, we can from the estimate-column in the summary-output find the estimates for the coeffiecients, the $\hat{\beta}$'s, belonging to each covariate. Thus, we can see that

$$\hat{\beta}_{BMI} = 3.087 \cdot 10^{-4}.$$

- Therefore, according to the model, if the BMI is increased by 1 while the other covariates are fixed, the response $-\frac{1}{\sqrt{SYSBP}}$ is estimated to increase by $3.087 \cdot 10^{-4}$.
- We have that for each β_j, $\hat{\beta}_j \sim N(\beta_j, \text{Var}(\hat{\beta}_j))$ and

$$T_j = \frac{\hat{\beta}_j - \beta_j}{\sqrt{\widehat{\text{Var}}(\hat{\beta}_j)}} \sim t_{n-2}.$$

  Using this, we can construct a confidence interval for $\hat{\beta}_{BMI}$. We have that

$$P(-t_{\alpha/2,n-2} < \frac{\hat{\beta}_j - \beta_j}{\sqrt{\widehat{\text{Var}}(\hat{\beta}_j)}} < t_{\alpha/2,n-2}) = 1 - \alpha.$$

  From this, we will get an interval on the form $\hat{\beta}_{BMI} \pm t_{\alpha/2,n-2} SE(\hat{\beta}_{BMI})$. In order to make a 99 % confidence interval, we have $\alpha = 0.01$. Thus, we can look up in a table to find the value $t_{0,005,n-2} \approx z_{0,005} = 2.58$. Since $n = 2600 > 30$, we can use this approximation by the Central Limit Theorem. Now, by using the values of $\hat{\beta}_{BMI}$ and $SE(\hat{\beta}_{BMI})$ from the summary-output, we obtain the 99 % confidence interval $(3.087 \cdot 10^{-4} \pm 2.58 * 2.955 \cdot 10^{-5})$,

$$(2.325 \cdot 10^{-4}, 3.849 \cdot 10^{-4}).$$

- The interpretation of this interval is that we can have 99 % confidence that the true β is within this interval. Considering the hypothesis test $H_0 : \beta_{BMI} = 0$ against $H_1 : \beta_{BMI} \neq 0$, we can get information about the p-value for this test through the confidence interval. Since the confidence interval does not contain the null hypothesis value $\beta_{BMI} = 0$, we know that the p-value is less than the significance level $\alpha = 0.01$. Equivalently, the hypothesis test is statistically significant.

d) Prediction

Now we consider a person with these data:

```
names(data)
```

```
## [1] "SYSBP"    "SEX"      "AGE"      "CURSMOKE" "BMI"      "TOTCHOL"
## [7] "BPMEDS"
```

```
new=data.frame(SEX=1,AGE=56,CURSMOKE=1,BMI=89/1.75^2,TOTCHOL=200,BPMEDS=0)
```

- In order to make a guess for his $-\frac{1}{\sqrt{SYSBP}}$, we insert these data into the equation for the multiple linear regression modelA, with the estimated values $\hat{\beta}_j$ from task a. We then get the response value -0.08667, which is our best guess. If we have that $y = -\frac{1}{\sqrt{SYSBP}}$, we have that the inverse function is equal to $SYSBP = \frac{1}{y^2}$. Thus, when using that the best guess for y is -0.08667, we get that the best guess for the systolic blood pressure of this person is $SYSBP = \frac{1}{(-0.08667)^2} = 133.1$.

- To make a 90 % prediction interval for this person's systolic blood pressure SYSBP, we can construct a prediction interval for the response of `modelA` around our best guess, and then transform the limits of the interval by the inverse function of $-\frac{1}{\sqrt{SYSBP}}$:

```
predict(modelA,newdata=new,interval ="prediction",type="response",level=0.90)
```

```
##           fit         lwr         upr
## 1 -0.08667246 -0.09625664 -0.07708829
```

```
"Limits of prediction interval for SYSBP:"
```

```
## [1] "Limits of prediction interval for SYSBP:"
```

```
lower = 1/(predict(modelA,newdata=new,interval ="prediction",type="response",level=0.90)[2]^2)
lower
```

```
## [1] 107.9291
```

```
upper = 1/(predict(modelA,newdata=new,interval ="prediction",type="response",level=0.90)[3]^2)
upper
```

```
## [1] 168.2764
```

- Thus, according to our model, the probability is 90 % that this person has a systolic blood pressure between 107.9 and 168.3. The range of this interval is very large, so the prediction interval is not that informative. If we look at the dataset, we can see that a very large amount of the people in the study has a systolic blood pressure in this interval. Therefore, we would have wanted a smaller range of the prediction interval in order to get useful information from it.

Problem 3 - Classification

a) Logistic regression

- We want to show that $logit(p_i) = log(\frac{p_i}{1-p_i})$ is a linear function, where $p_i = \frac{exp(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2})}{1+exp(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2})}$. Thus, we get:

$$log\Big(\frac{p_i}{1-p_i}\Big) = log\left(\frac{\frac{exp(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2})}{1+exp(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2})}}{1-\frac{exp(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2})}{1+exp(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2})}}\right) = log\left(\frac{\frac{exp(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2})}{1+exp(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2})}}{\frac{1}{1+exp(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2})}}\right)$$
$$= log\Big(exp(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2})\Big) = \beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}.$$

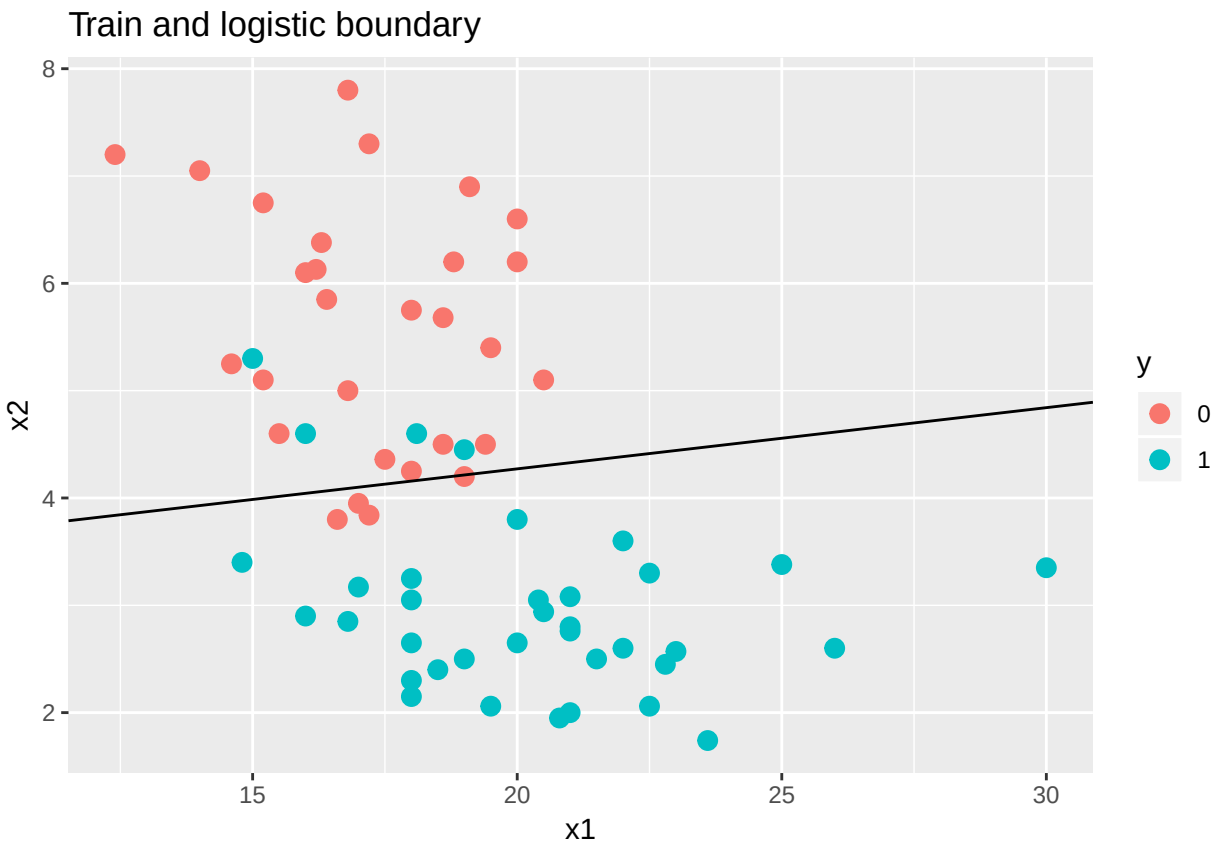
  And we get $log\Big(\frac{p_i}{1-p_i}\Big) = \beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}$, which is a linear function.

```
## Warning: package 'pROC' was built under R version 3.4.4
```

```
## (Intercept)          x1          x2
##   7.3143074   0.1332006  -2.3360758
```

- There is no obivous interpretation of the $\hat{\beta}_1$ and $\hat{\beta}_2$ since an increase of one unit in $\hat{\beta}$ does not give a consistent increase or decrease for all values of x_1. It is however easier to look at the odds $\frac{p_i}{1-p_i}$, which is bounded from below, but not above. The interpretation of the odds in this example is the conditional probability that a wine is in class 1, divided by the probability that the wine is in class 0 for the covariates x_1 and x_2. Moreover, if there is an increase from x_{1i} to $x_{1i}+1$, the odds are multiplied by e^{β_1}, which gives us a more intuitive interpretation of $\hat{\beta}_1$. The case is similar for $\hat{\beta}_2$ for an increase in x_{2i}, and we can see that for $\beta < 0$ the odds will decrease, for $\beta_1 = 0$ the odds will stay the same, and for $\beta_1 > 0$, the odds will increase.

- Since the odds is linear (as shown before), we have a linear class boundary. From earlier, we have: $log\left(\frac{Pr(Y_i=1|X=x_i)}{Pr(Y_i=0|X=x_i)}\right) = log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}$, where we want to fit the coefficients by maximizing the likelihood. Then, where $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} = 0$, we have the class boundary.




```
##
## Call:
## glm(formula = y ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.46217  -0.17536   0.09309   0.28590   2.49572
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   7.3143     5.3382   1.370 0.170626
## x1            0.1332     0.2194   0.607 0.543800
## x2           -2.3361     0.6472  -3.609 0.000307 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 89.354  on 64  degrees of freedom
## Residual deviance: 30.027  on 62  degrees of freedom
## AIC: 36.027
##
```

```
## Number of Fisher Scoring iterations: 6
```

- Using the summary output, we can get the coefficients β_0, β_1 and β_2. Thus by using the formula:

$$P(Y = 1|x_1 = 17, x_2 = 3) = \frac{exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2)}{1 + exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2)} = \frac{exp(7.3143 + 0.1332 * 17 - 2.3361 * 3)}{1 + exp(7.3143 + 0.1332 * 17 - 2.3361 * 3)}$$
$$= \frac{exp(2.5704)}{1 + exp(2.5704)} = 0.9289.$$

  This result can be interpreted as the probability for a observation to be in class $Y = 1$ given the variable values $x_1 = 17$ and $x_2 = 3$.

- The sensitivity is the proportion of correctly classified positive observations, and the specificity is the proportion of correctly classified negative observations. Thus, the values are between 0 and 1, and the goal is to obtain high values.

```
## Warning: package 'knitr' was built under R version 3.4.4
```

	Predicted -	Predicted +	Total
True -	25	5	30
True +	5	30	35
Total	30	35	

- The tables above shows the confusion table for the test data, based on the from the training data. It is important to note that a large value for only one of them is not necessarily any good. For example, a method which just puts every observation in class 1 will have sensitivity 1, but specificity 0. Therefore, we want high values (close to one), but balanced as well. There are cases where it is more important that we get the classification of one class right than the other, for which we can sacrifice the balance to obtain either a high sensitivity or a high specificity (example would be to avoid a Type 1 error, for which we can sacrifice the probability for actually rejecting a false H_0). That being said, in this case there is no worse to classify a 1-wine as a 0-wine, than the opposite.

```
## [1] 0.8571429
```

```
## [1] 0.8333333
```

They both obtain relatively large values, and one can therefore argue that the classification works pretty good. Similar values for the sensitivity and the specificity suggest that the classification does not prefer or default to one class, which in this case as discussed above is desirable.

b) K-nearest neighbor classifier

$$Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_o} I(y_i = j)$$

- Given the point x_0, the equation finds the K nearest points in the dataset, and count the most frequent class among the neighbours. The right hand side of the equation finds the percentage of the neighbours which belong to class j, and based on that, we take a decision whether $x_0 =$ should be in class j or not.

```
##    testclass3
##       0  1
##   0 25  5
##   1  3 32
```

```
## [1] 0.9142857
```

```
## [1] 0.8333333
```

- Here, the sensitivity is a little higher than the logistic regression, and the specificity stays the same. Thus, this method (with K=3) gets a total higher number of wines classified from the right cultivar. This is a good indicator that KNN with K=3 performs better in this example than the logistic regression.

```
##    testclass9
##       0  1
##   0 25  5
##   1  5 30
```

```
## [1] 0.8571429
```

```
## [1] 0.8333333
```

- As we can see from the data, the sensitivity for K=3 is higher than for K=9, and the specificity is equal. That happens because the neighbours in K=9 becomes further away, and even though the formula takes account for more points, the points are not as local anymore. Thus the bias increases, and the misclassification increases as well. The best value for K really depends on the number of points and how they are distributed, but as shown in 3d), one can calculate the ROC curve, and find the optimal K. Since there is a bias-variance trade-off, where low values for K will give high variance and high values for K will give a high bias, the optimal point will be where they together give the lowest mean squared error (MSE).

c) LDA (& QDA)

The expression from the task is found by assuming the classes are normally distributed according to the function

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_k})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_k})}.$$

The posterior probability is the calculated as

$$p_k(x) = Pr(Y = k|\boldsymbol{X} = \boldsymbol{x}) = \frac{Pr(\boldsymbol{X} = \boldsymbol{x}|Y = k)Pr(Y = k)}{Pr(\boldsymbol{X} = \boldsymbol{x})} = \frac{\pi_k f_k(\boldsymbol{x})}{\sum_{l=1}^{K} \pi_l f_l(\boldsymbol{x})}$$

- Here, $\pi_k = Pr(Y = k)$, which is the probability that a given wine sample Y belongs to class k. $\boldsymbol{\mu_k}$ is the vector containing the expected values of the parameters, which is color intensity and alcalinity of ash in this task. $\boldsymbol{\Sigma}$ is the covariance matrix for the parameters, which contains the variance of the parameters and the covariance between them. $f_k(x)$ is as mentioned the probability density function for a given class, which again is the conditional probability of x with a given class, $f_k(x) = Pr(\boldsymbol{X} = \boldsymbol{x}|Y = k)$.
- To estimate π_k we can use the total number of samples in the training set n, and the number of samples belonging to the requested class, n_k. We can then use the estimator $\hat{\pi}_k = \frac{n_k}{n}$. μ_k can be estimated simply using a sample mean from all the samples belonging to the requested class. The estimator is then calculated as $\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$. To estimate the covariance matrix, the class-specific covariance matrices can be calculated as

  $$\hat{\boldsymbol{\Sigma}}_k = \frac{1}{n_k - 1} \sum_{i:y_i=k} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)^T.$$

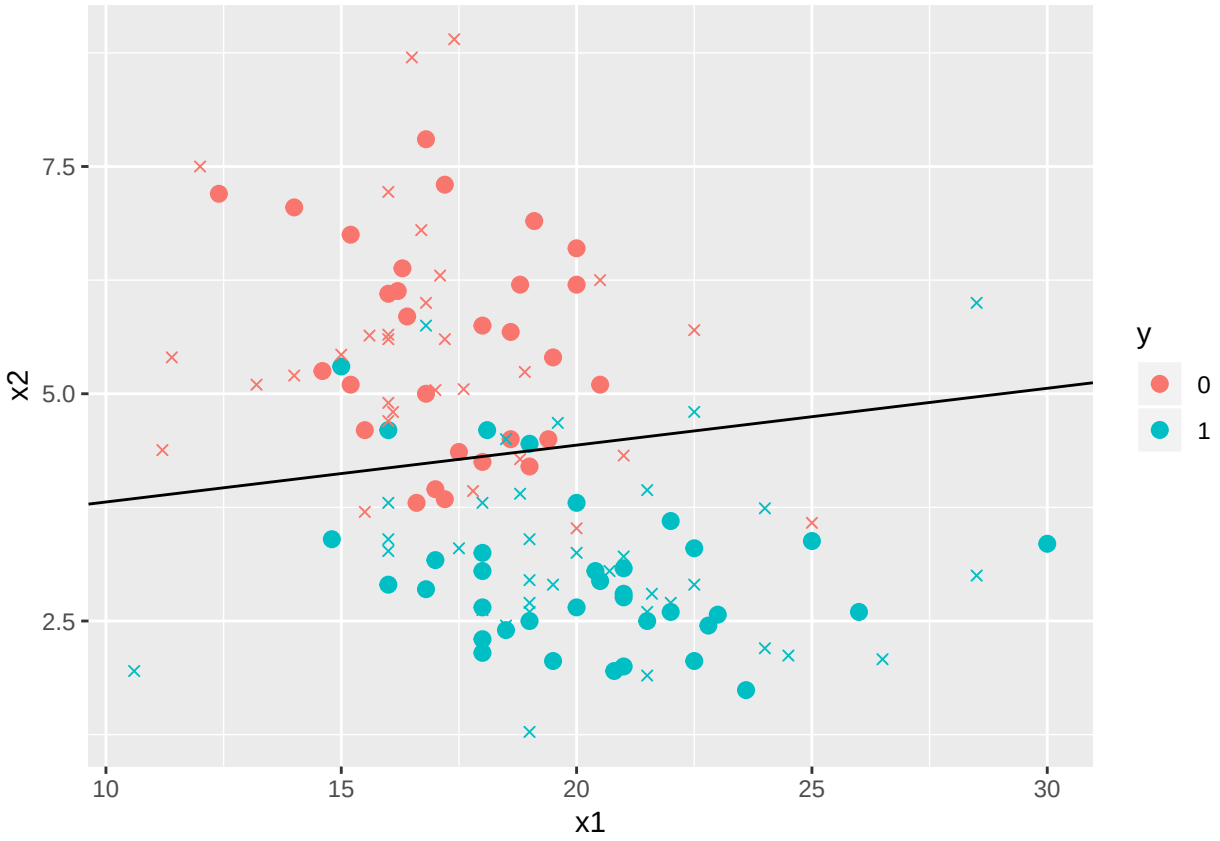
  The true covariance matrix is then found by summing over the different classes using weights according to the fraction of samples in the belonging class. We then get

  $$\hat{\boldsymbol{\Sigma}} = \sum_{k=1}^{K} \frac{n_k - 1}{n - K} \cdot \hat{\boldsymbol{\Sigma}}_k.$$

Calculations for the training data is done in R, resulting in values $\pi_0 = 0.45$, $\pi_1 = 0.55$, $\boldsymbol{\mu_0} = (17.26, 5.58)^T$, $\boldsymbol{\mu_1} = (20.18, 2.97)^T$ and

$$\hat{\boldsymbol{\Sigma}} = \begin{pmatrix} 7.23 & -0.66 \\ -0.66 & 0.95 \end{pmatrix}$$

- We want to find when
$$Pr(Y = 0|\boldsymbol{X} = \boldsymbol{x}) = Pr(Y = 1|\boldsymbol{X} = \boldsymbol{x})$$
We rewrite this as
$$\frac{\pi_0 e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_0})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_0})}}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}\sum_{l=1}^{K}\pi_l f_l(\boldsymbol{x})} = \frac{\pi_1 e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_1})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_1})}}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}\sum_{l=1}^{K}\pi_l f_l(\boldsymbol{x})}$$
Since the denominators are equal on both sides, we can disregard them. By taking logarithms on both sides we achieve
$$-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_0})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_0}) + \log\pi_0 = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_1})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_1}) + \log\pi_1$$
We now complete the multiplications, and remove the equal parts on each side to get:
$$\frac{1}{2}(\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_0} + \boldsymbol{\mu_0}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \boldsymbol{\mu_0}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_0}) + \log\pi_0 = \frac{1}{2}(\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_1} + \boldsymbol{\mu_1}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \boldsymbol{\mu_1}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_1}) + \log\pi_1$$
Now we know the covariance matrix is symmetric positive definite. We therefore know $\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_k} = \boldsymbol{\mu_k}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}$. We therefore finally get
$$\delta_0(\boldsymbol{x}) = \delta_1(\boldsymbol{x}),$$
where
$$\delta_k(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_k} - \frac{1}{2}\boldsymbol{\mu_k}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_k} + \log\pi_k; \quad k \in \{0, 1\}$$
.
- To find the class boundary formula, we need to solve the equation $\delta_0(\boldsymbol{x}) = \delta_1(\boldsymbol{x})$. We get
$$\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_0} - \frac{1}{2}\boldsymbol{\mu_0}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_0} + \log\pi_0 = \boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_1} - \frac{1}{2}\boldsymbol{\mu_1}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_1} + \log\pi_1$$
$$\boldsymbol{x}^T(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_0} - \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_1}) = \frac{1}{2}\boldsymbol{\mu_0}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_0} - \frac{1}{2}\boldsymbol{\mu_1}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_1} + \log\frac{\pi_1}{\pi_0}$$
We use estimators for the parameters as discussed earlier in this task. The necessary values are calculated in R, and we get $\log\frac{\pi_1}{\pi_0} = 0.216$, $\frac{1}{2}\boldsymbol{\mu_1}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_1} = -41.1$, $\frac{1}{2}\boldsymbol{\mu_0}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_0} = -49.2$ and $(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_0} - \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_1}) = (-0.17, 2.63)^T$. We then get the equation
$$2.628x_2 - 0.164x_1 = 8.368 \implies x_2 = 0.0625x_1 + 3.184$$
- The following plot shows the training data as circular dots, together with the test data as crosses. The plot also shows the line representing equal probability of being in the two classes, based on the training data using LDA.



- LDA is also performed using the built in R-function. This is done in the R-code shown below. We see the data corresponds to the estimators we have used earlier, which is natural as the estimators are calculated the same way.

```
lda_train=lda(y~x1+x2, data=train)
lda_train
```

```
## Call:
## lda(y ~ x1 + x2, data = train)
##
## Prior probabilities of groups:
##         0         1
## 0.4461538 0.5538462
##
## Group means:
##         x1       x2
## 0 17.25517 5.577241
## 1 20.17500 2.966944
##
## Coefficients of linear discriminants:
##            LD1
## x1  0.06067097
## x2 -0.97009878
```

	Predicted -	Predicted +	Total
True -	24	6	30
True +	5	30	35
Total	29	36	

- The table above shows the confusion table for the test data, based on the LDA from the training data. The sensitivity is given by

$$\frac{\text{True Positive}}{\text{Condition Positive}} = \frac{24}{30} = 0.8,$$

while the specificity is given by

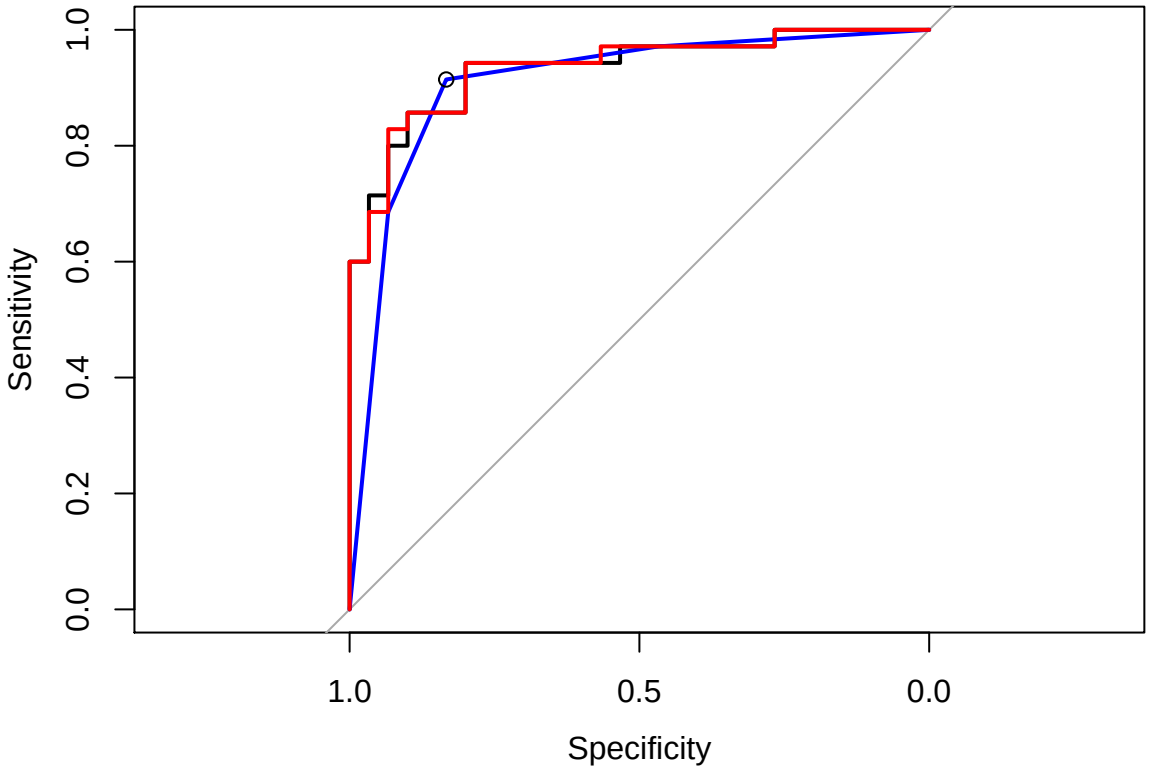
$$\frac{\text{True Negative}}{\text{Condition Negative}} = \frac{30}{35} = 0.86.$$

This result is quite equal to the earlier results from logistic regression and KNN, and they all seem to perform quite well as the numbers are close to 1.

- In QDA, the covariance matrices for the different classes are allowed to be different. This makes this method more complex, but also more flexible. The decision boundaries will now become quadratic functions of $\boldsymbol{x}$.

d) Compare classifiers

- For logistic regression we got specificity 0.833 and sensitivity 0.857. For KNN with our preferred $K = 3$, we got specificity 0.833 and sensitivity 0.914. For LDA we got specificity 0.857 and sensitivity 0.800. Since we want high specificity and sensitivity, our preferred method when using 0.5 as cut-off is KNN, since this has a better sensitivity, while the specificity is quite similar for all three methods. This method performs therefore somewhat better than the other two methods, which are more equal, though logistic regression seems to be slightly better. Our result is still an indication that a strictly linear model might not be optimal for the wine classification problem.
- The following plot show ROC curves for specificity and sensitivity, and also prints the area under the curves. From the last point we concluded that KNN was the best method for cut-off equal to 0.5. We do however see that the area under the curve is smaller for KNN than for the other methods. We also see on the plot the reason for this, as the slope in the lower left corner is less vertical, and the slope in the upper right corner is less horizontal. It therefore seems like the other methods would work better for high or low cut-off, compared to KNN. Logistic regression and LDA performs quite similar, which is natural as they make the both linearity-assumption. Lda has the highest area under the curve, and this might therefore be our method of choice for other cut-offs than 0.5.



* In this plot, black line is the ROC for logistic regression, blue line is the KNN (with K=3) and red line is the LDA. The AUC's printed in the same order below:

```
## Area under the curve: 0.9333
```

```
## Area under the curve: 0.9086
```

```
## Area under the curve: 0.9343
```