Extra exercises for STK2100

Geir Storvik

Spring 2018

Exercise 1 (Linear regression)
Consider the standard linear regression model

$$Y_i = \boldsymbol{x}_i\boldsymbol{\beta} + \varepsilon_i, \quad i = 1, ..., n$$

where $\varepsilon_1, ..., \varepsilon_n \overset{iid}{\sim} N(0, \sigma^2)$. We can also write the model more compact as

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}_i$$

where $\boldsymbol{X}$ here is a matrix with row i corresponding to $\boldsymbol{x}_i$ and $\boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \sigma^2\boldsymbol{I}_n)$ with $\boldsymbol{I}_n$ being the identity matrix of dimension $n \times n$. We will assume $\boldsymbol{X}$ known in this exercise.

(a) Show that the maximum likelihood estimator for $\boldsymbol{\beta}$ is equal to the least squares estimator

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmin}}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}).$$

(b) Show that the maximum likelihood estimator for $\boldsymbol{\beta}$ is

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

(c) Find the maximum likelihood estimator for σ^2.

Consider a random vector $\boldsymbol{Z} = (Z_1, ..., Z_q)$. The *expectation vector* $\boldsymbol{\mu}_z$ for $\boldsymbol{Z}$, $\mathsf{E}(\boldsymbol{Z})$, is a vector with ith element equal to the expectation of Z_i. The *covariance matrix* for a random vector $\boldsymbol{Z}$, $\mathsf{V}(\boldsymbol{Z})$, is a matrix where the (i, j) element is equal to the covariance between Z_i and Z_j.

(d) Show that if $\boldsymbol{A}$ is a matrix and $\boldsymbol{b}$ a vector, then $\mathsf{E}(\boldsymbol{A}\boldsymbol{Z} + \boldsymbol{b}) = \boldsymbol{A}\mathsf{E}(\boldsymbol{Z}) + \boldsymbol{b}$ and that $\mathsf{V}(\boldsymbol{A}\boldsymbol{Z} + \boldsymbol{b}) = \boldsymbol{A}\mathsf{V}(\boldsymbol{Z})\boldsymbol{A}^T$.

What requirements are there on the matrix $\boldsymbol{A}$ and the vector $\boldsymbol{b}$?

(e) Show that a covariance matrix always needs to be positive (semi-)definite.

Hint: A matrix $\boldsymbol{C}$ is positive semi-definite if for any vector $\boldsymbol{a}$ we have $\boldsymbol{a}^T\boldsymbol{C}\boldsymbol{a} \geq 0$.

(f) Show that the expectation vector of $\boldsymbol{Y}$ is $\boldsymbol{X}\boldsymbol{\beta}$ and use this to show that the expectation vector of $\widehat{\boldsymbol{\beta}}$ is equal to $\boldsymbol{\beta}$.

(g) Show that the covariance matrix for $\widehat{\boldsymbol{\beta}}$ is equal to $\sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}$.

Assume now that the first column of $\boldsymbol{X}$ contain only 1's (corresponding to an intercept term) while the rest of the elements of $\boldsymbol{X}$ are generated according to the $N(0,1)$ distribution.

(h) Within R, generate $\boldsymbol{X}$ for $p = 5$ and $n = 10, 15, 20, 25, \cdots, 95, 100$ and plot the variance of the first component of $\widehat{\boldsymbol{\beta}}$ as a function of n. Make your own choice of σ^2.

(i) Within R, generate $\boldsymbol{X}$ for $n = 31$ and $p = 20, 21, ..., 30, 31, 32$ and plot the variance of the first component of $\widehat{\boldsymbol{\beta}}$ as a function of p. Make your own choice of σ^2.

(j) Discuss these results.

Exercise 2 (Statistical decision theory)
Assume a setting where we have explanatory variables $\boldsymbol{x} \in \boldsymbol{R}^p$ and a response variable $Y \in \boldsymbol{R}$. We want to find a predictor $f(\boldsymbol{x})$ for prediction of Y.

Evaluation of how good $f(\boldsymbol{x})$ is for prediction can be measured by a *loss function* $L(Y, f(\boldsymbol{x}))$. Statistical desisions theory can be used to derive the optimal predictor given a specific loss function. In this exercise, we will look at one of the most commonly used loss functions, the *squared loss function*

$$L(y, f(\boldsymbol{x})) = (y - f(\boldsymbol{x}))^2$$

In principle, we want to minimize $L(y, f(\boldsymbol{x}))$. However, when applying the predictor, y will be unknown and the loss is therefore not possible to evaluate. An alternative is then to try to minimize the *expected loss* or *expected prediction error* (EPE) which is given by

$$\text{EPE}(f) = \mathsf{E}(L(Y, f(\boldsymbol{X}))) = \mathsf{E}(Y - f(\boldsymbol{X}))^2 = \int_{\boldsymbol{x},y} (y - f(\boldsymbol{x}))^2 p(\boldsymbol{x}, y) dy d\boldsymbol{x}$$

where $p(\boldsymbol{x}, y)$ is the density for $(\boldsymbol{X}, Y)$. Our task will be to find the function $f(\boldsymbol{x})$ which minimizes EPE(f). Note that we in this case also are considering the explanatory variables as random variables.

(a) Show that f is the function given by

$$f(\boldsymbol{x}) = \underset{c}{\text{argmin}}\, \mathsf{E}_{Y|\boldsymbol{X}=\boldsymbol{x}}([Y - c]^2 | \boldsymbol{X} = \boldsymbol{x})$$

Hint: Use that $\mathsf{E}(Y - f(\boldsymbol{X}))^2 = \mathsf{E}_{\boldsymbol{X}}\mathsf{E}_{Y|\boldsymbol{X}}([Y - f(\boldsymbol{x})]^2 | \boldsymbol{X} = \boldsymbol{x})$

(b) Show that the value c than minimizes $\mathsf{E}_{Y|\boldsymbol{x}}([y - c]^2 | \boldsymbol{X} = \boldsymbol{x})$ is

$$c = \mathsf{E}[Y | \boldsymbol{X} = \boldsymbol{x}].$$

(*c*) Consider now a model

$$Y = g(\boldsymbol{X}) + \varepsilon$$

where ε is a random variable with zero expectation. What is the optimal predictor in this case?

(*d*) Assume the model above and further that $\mathsf{V}(\varepsilon) = \sigma^2$ and that ε is independent of $\boldsymbol{X}$. Consider a general predictor $f(\boldsymbol{x})$ (which might be different from the optimal choice that you derived in the previous point). Derive a general expression for $\text{EPE}(f)$ and show that a lower limit is $\mathsf{V}(g(\boldsymbol{X})) + \sigma^2$.

Exercise 3 (Loss functions for binary responses)
Assume now a setting where we want to predict a binary response $Y \in \{0, 1\}$ based on some explanatory variables $\boldsymbol{X} \in \boldsymbol{R}^p$. Assume that the predictor $f(\boldsymbol{x}) \in \{0, 1\}$. A possible loss function in this case is

$$L(y, f(\boldsymbol{x})) = \begin{cases} 0 & \text{if } f(\boldsymbol{x}) = y; \\ 1 & \text{otherwise.} \end{cases}$$

Also in this case we want to minimize

$$\text{EPE}(f) = \mathsf{E}(L(Y, f(\boldsymbol{X}))$$

(*a*) Using similar arguments as for exercise 2, show that

$$\text{EPE}(f) = \int_{\boldsymbol{x}} [1 - \Pr(Y = f(\boldsymbol{x}) | \boldsymbol{X} = \boldsymbol{x})] p(\boldsymbol{x}) d\boldsymbol{x}$$

where $p(\boldsymbol{x})$ is the marginal density of $\boldsymbol{X}$ and $\Pr(Y \neq f(\boldsymbol{x}) | \boldsymbol{X} = \boldsymbol{x})$ is the conditional probability of Y being different of the predictor for a given value of $\boldsymbol{x}$.

(*b*) Show that the optimal predictor is

$$f(\boldsymbol{x}) = \underset{k \in \{0,1\}}{\text{argmin}} [1 - \Pr(Y = k | \boldsymbol{x})] = \underset{k \in \{0,1\}}{\text{argmax}} \Pr(Y = k | \boldsymbol{x})$$

and that this corresponds to

$$f(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \Pr(Y = 1 | \boldsymbol{x}) \geq 0.5; \\ 0 & \text{otherwise.} \end{cases}$$

Discuss this result.

(*c*) Extend the results above to the case where $Y \in \{0, ..., K - 1\}$.

(*d*) Argue why the expected error rate for $\boldsymbol{X} = \boldsymbol{x}$ will be $1 - \max_k \Pr(Y = k | \boldsymbol{x})$.

Exercise 4 (Linear regression)
We will in this exercise look at different aspects related to linear regression. Consider first a very simple situation where the *true* model is

$$Y = \beta_0 + \beta_1 x + \varepsilon, \quad \varepsilon \overset{iid}{\sim} N(0, \sigma^2). \tag{*}$$

Assuming we have available data $\{(x_1, y_1), ..., (x_n, y_n)\}$, we can *test* the hypothesis

$$H_0 : \beta_1 = 0 \text{ against } H_a : \beta_1 \neq 0$$

by using that under H_0

$$T = \frac{\hat{\beta}_1}{\text{SE}(\hat{\beta}_1)} \sim t_{n-2}.$$

We will look at the performance of this test through a simulation study.

(a) Assume $n = 30, \beta_0 = 1, \sigma^2 = 1$. For $\beta_1 = 0.0, 0.1, 0.2, \cdots, 1.9, 2.0$, simulate 100 datasets where $X \sim N(0, 1)$ and Y follows model (*). Record for each simulation whether H_0 is rejected or not (using a signficance level $\alpha = 0.05$) and plot the rejection rate as a function of β_1.

Discuss the results.

Hint: At the course webpage there is an R script `extra4.r` which you can use for this task. Make however sure that you understand what is going on.

Assume now that we also have available a second explanatory variable z. We will also assume that these variables are generated according to $Z \sim N(0, 1)$ but that $\text{Cor}(X, Z) = 0.9$. We will however still assume that (*) is the *true* model but that we in our analysis *assume*

$$Y = \beta_0 + \beta_1 x + \beta_2 z + \varepsilon, \quad \varepsilon \overset{iid}{\sim} N(0, \sigma^2). \tag{**}$$

(b) Show that if

$$Z = 0.9X + \sqrt{1 - 0.9^2}\eta$$

where $\eta \sim N(0, 1)$ and independent of X then (X, Z) will have the properties above. Explain how this can be used to simulate the pair (X, Z) on the computer.

(c) Modify your script from (a) so that you simulate values of the triplets $\{(x_i, z_i, y_i), i = 1, ..., n\}$ from the true model, but now fit model (**). Record and plot the rejection rate of H_0 as a function of β_1 also in this case.

(d) What happends if you increase the correlation between X and Z to 0.99?

(*e*) Discuss these results. Are there some aspects that have not been taken into account in these simulation experiments?

Exercise 5 (Prediction)
Assume the standard linear regression model

$$Y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon = \boldsymbol{x}^T \boldsymbol{\beta} + \varepsilon$$

with the usual assumptions about the noise terms. We will in this exercise consider prediction at a new point $\boldsymbol{x}^*$,

$$Y^* = \beta_0 + \beta_1 x_1^* + \cdots + \beta_p x_p^* + \varepsilon^* = (\boldsymbol{x}^*)^T \boldsymbol{\beta} + \varepsilon^*$$

We will denote the ordinary least squares estimate for $\boldsymbol{\beta}$ by $\widehat{\boldsymbol{\beta}}$ as usual.

(*a*) Denote $\theta = (\boldsymbol{x}^*)^T \boldsymbol{\beta}$. Show that $\hat{\theta} = (\boldsymbol{x}^*)^T \widehat{\boldsymbol{\beta}}$ is an unbiased estimate of θ.

(*b*) Derive an expression for the variance of $\hat{\theta}$, $\sigma^2_{\hat{\theta}}$.

You may in the following use that

$$(n-p-1)\frac{\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{n-p-1},$$

that $\hat{\boldsymbol{\beta}}$ is independent of $\hat{\sigma}^2$ and that if $Z \sim N(0,1)$ and $X \sim \chi^2_\nu$ with Z and X independent, then

$$T = \frac{Z}{\sqrt{X/\nu}} \sim t_\nu.$$

(Here χ^2_ν is the Chi-square distribution with ν degrees of freedom and t_ν is the t distribution with ν degrees of freedom.)

(*c*) Argue why

$$T = \frac{\hat{\theta} - \theta}{s_{\hat{\theta}}} \sim t_{n-p-1}$$

where $s_{\hat{\theta}}$ is the estimate of $\sigma_{\hat{\theta}}$ with $\hat{\sigma}$ is inserted for σ.

Use this to construct a $100(1-\alpha)\%$ confidence interval for θ.

(*d*) Consider now prediction of Y^*. Show that $E[Y^* - \hat{\theta}] = 0$.

Why do we not state this as $E[\hat{\theta}] = Y^*$?

(*e*) Derive the variance of $Y^* - \hat{\theta}$, $\sigma^2_{Y^* - \hat{\theta}}$.

(f) Argue why

$$T = \frac{Y^* - \hat{\theta}}{s_{Y^*-\hat{\theta}}} \sim t_{n-p-1}$$

where $s_{Y^*-\hat{\theta}}$ is the estimate of $\sigma_{Y^*-\hat{\theta}}$ with $\hat{\sigma}$ is inserted for σ.

Use this to construct a $100(1-\alpha)\%$ prediction interval for Y^*.

(g) Consider now the `Advertising` data from the text book (with an **R**-script available from the course web-page). In the **R**-script a command is included that make the prediction interval. Use `help(predict.lm)` to see how you can construct a confidence interval for Y^*. Discuss the similarities and the difference between this interval and the prediction interval.

Why do you need to use `help(predict.lm)` and not `help(predict)` here?

Exercise 6 (More general loss functons for binary responses)
Consider again the setting of Exercise 3. We will however now consider a more general loss function

$$L(y, f(\boldsymbol{x})) = \begin{cases} 0 & \text{if } f(\boldsymbol{x}) = y; \\ c_0 & \text{if } y = 0 \text{ and } f(\boldsymbol{x}) = 1; \\ c_1 & \text{if } y = 1 \text{ and } f(\boldsymbol{x}) = 0; \end{cases}$$

indicating that we put different losses on the two types of errors.

We continue to minimize

$$\text{EPE}(f) = \mathsf{E}(L(Y, f(\boldsymbol{X}))$$

(a) Show that

$$\begin{aligned} \text{EPE}(f) &= \int_{\boldsymbol{x}; f(\boldsymbol{x})=0} Q_1(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} + \int_{\boldsymbol{x}; f(\boldsymbol{x})=1} Q_0(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} \\ &= \int I(f(\boldsymbol{x}) = 0) Q_1(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} + \int I(f(\boldsymbol{x}) = 1) Q_0(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} \end{aligned}$$

where

$$\begin{aligned} Q_0(\boldsymbol{x}) &= c_0 \Pr(Y = 0 | \boldsymbol{X} = \boldsymbol{x}) \\ Q_1(\boldsymbol{x}) &= c_1 \Pr(Y = 1 | \boldsymbol{X} = \boldsymbol{x}) \end{aligned}$$

and $I(\cdot)$ is the usual indicator function.

(b) Show that the expression above can be rewritten to

$$\text{EPE}(f) = \text{Const} + \int I(f(\boldsymbol{x}) = 0)[Q_1(\boldsymbol{x}) - Q_0(\boldsymbol{x})] p(\boldsymbol{x}) d\boldsymbol{x}$$

and use this to argue that the optimal predictor in this case is

$$f(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \Pr(Y=1|\boldsymbol{X}=\boldsymbol{x}) > \frac{c_0}{c_1}\Pr(Y=0|\boldsymbol{X}=\boldsymbol{x}); \\ 0 & \text{otherwise.} \end{cases}$$

Discuss why this is a reasonable classification rule in this case.

Exercise 7 (Leave-one-out cross-validation and linear regression)
We will in this case explore the LOOCV procedure in the case of the linear regression model

$$\begin{aligned} Y_i =& \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p} + \varepsilon_i \\ =& \boldsymbol{x}_i^T \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, ..., n \end{aligned}$$

where $\boldsymbol{x}_i = (1, x_{i,1}, ..., x_{i,p})^T$ and $\boldsymbol{\beta} = (\beta_0, ..., \beta_p)$. We recall that the least squares estimate is given by

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

while the *hat matrix* is defined by

$$\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T.$$

In order to make clear that there are different types of predictions, we will use $\hat{y}_i$ for the prediction of y_i based on *all* the data while we will use $\hat{y}_i^{-i}$ to be the prediction of y_i when observation i is excluded (the leave-one-out prediction).

(a) We will start with the case $p = 0$.

(i) Show that $\hat{y}_i = \bar{y}$ for all i in this case.

(ii) Show that the LOOCV prediction of y_i in this case is $\hat{y}_i^{-i} = \frac{1}{n-1}\sum_{j\neq i} y_j$.

(iii) Show that $h_{ii} = \frac{1}{n}$ in this case.

(iv) Show that

$$y_i - \hat{y}_i^{-i} = \frac{y_i - \hat{y}_i}{1 - \frac{1}{n}}$$

and that this corresponds to equation (5.2) in the textbook.

(b) Consider now the general case $p \geq 1$. We will start by consider the special case $i = n$. We will also denote by $\boldsymbol{X}_s$ the design matrix based on the first s observations and similarly $\boldsymbol{y}_s$ to be the vector of the first s responses. Note that $\boldsymbol{X} = \boldsymbol{X}_n$ and $\boldsymbol{y} = \boldsymbol{y}_n$. We will also use the notation $\boldsymbol{M}_n = \boldsymbol{X}_n^T\boldsymbol{X}_n$.

(i) Show that $\boldsymbol{M}_n = \sum_{i=1}^n \boldsymbol{x}_i\boldsymbol{x}_i^T$.

(*ii*) Show in general that if $\boldsymbol{A}$ is a symmetric non-singular matrix and $\boldsymbol{v}$ is a vector, then

$$[\boldsymbol{A}+\boldsymbol{u}\boldsymbol{v}^T]^{-1}=\boldsymbol{A}^{-1}-\tfrac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^T\boldsymbol{A}^{-1}}{1+\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u}}.$$

This is known as the Sherman-Morrison formula within the linear algebra theory.

(*iii*) Show that

$$\boldsymbol{M}_n^{-1}=\boldsymbol{M}_{n-1}^{-1}-\tfrac{\tilde{\boldsymbol{x}}_n\boldsymbol{x}_n^T\boldsymbol{M}_{n-1}^{-1}}{1+\boldsymbol{x}_n^T\tilde{\boldsymbol{x}}_n}$$

where

$$\tilde{\boldsymbol{x}}_n=\boldsymbol{M}_{n-1}^{-1}\boldsymbol{x}_n.$$

(*iv*) Denote by $\hat{\boldsymbol{\beta}}_s=\boldsymbol{M}_s^{-1}\boldsymbol{X}_s^T\boldsymbol{y}_s$ the least squares estimate based on the first s observations. Show that

$$\begin{aligned}\hat{\boldsymbol{\beta}}&=\hat{\boldsymbol{\beta}}_n\\&=\left[\boldsymbol{M}_{n-1}^{-1}-\tfrac{\tilde{\boldsymbol{x}}_n\boldsymbol{x}_n^T\boldsymbol{M}_{n-1}^{-1}}{1+\boldsymbol{x}_n^T\tilde{\boldsymbol{x}}_n}\right][\boldsymbol{X}_{n-1}^T\boldsymbol{y}_{n-1}+\boldsymbol{x}_n y_n]\\&=[\boldsymbol{I}-\tfrac{\tilde{\boldsymbol{x}}_n\boldsymbol{x}_n^T}{1+\boldsymbol{x}_n^T\tilde{\boldsymbol{x}}_n}]\hat{\boldsymbol{\beta}}_{n-1}+\tfrac{1}{1+\boldsymbol{x}_n^T\tilde{\boldsymbol{x}}_n}\tilde{\boldsymbol{x}}_n y_n.\end{aligned}$$

(*v*) Show that

$$[\boldsymbol{I}-\tfrac{\tilde{\boldsymbol{x}}_n\boldsymbol{x}_n^T}{1+\boldsymbol{x}_n^T\tilde{\boldsymbol{x}}_n}]^{-1}=\boldsymbol{I}+\tilde{\boldsymbol{x}}_n\boldsymbol{x}_n^T$$

and use this to show that

$$\hat{\boldsymbol{\beta}}_{n-1}=[\boldsymbol{I}+\tilde{\boldsymbol{x}}_n\boldsymbol{x}_n^T]\hat{\boldsymbol{\beta}}_n-\tilde{\boldsymbol{x}}_n y_n$$

(*vi*) Denoting now $\hat{y}_n^{-n}=\boldsymbol{x}_n^T\hat{\boldsymbol{\beta}}_{n-1}$, show that

$$y_n-\hat{y}_n^{-n}=(1+\boldsymbol{x}_n^T\tilde{\boldsymbol{x}}_n)(y_n-\hat{y}_n)$$

(*vii*) Show that the nth diagonal element of $\boldsymbol{H}$ can be written as

$$\begin{aligned}H_{n,n}&=\boldsymbol{x}_n^T\boldsymbol{M}_n^{-1}\boldsymbol{x}_n\\&=\tfrac{\boldsymbol{x}_n^T\tilde{\boldsymbol{x}}_n}{1+\boldsymbol{x}_n^T\tilde{\boldsymbol{x}}_n}\end{aligned}$$

and use this to verify equation (5.2) in the textbook for $i=n$

(*viii*) Use symmetry arguments to show that equation (5.2) is valid for all $i=1,...,n$.

(*c*) Discuss how these results can be used for recursive calculation of least squares estimates (as well as the related covariance matrix).

Exercise 8 (Constant constraints on cubic splines)
Assume we have a cubic spline model

$$Y = g(x) + \varepsilon$$

where

$$g(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{k=1}^{K} \beta_{3+k} b_{3+k}(x)$$

and

$$b_{3+k}(x) = (x - c_k)_+^3 \text{ for } k = 1, ..., K$$

Assume now that we want to impose the constraint that $g(x)$ is constant in the intervals $(-\infty, c_1)$ and $[c_K, \infty)$. We will see what kind of model this reduces to.

(a) Argue why this also can considered as a problem of imposing extra constraints on natural splines, given by

$$g(x) = \theta_0 + \theta_1 x + \sum_{k=1}^{K-2} \theta_{1+k} n_k(x)$$

w here

$$n_k(x) = d_k(x) - d_{K-1}(x), k = 1, ..., K - 2$$

and

$$d_k(x) = \frac{(x - c_k)_+^3 - (x - c_K)_+^3}{c_K - c_k}$$

(b) Show that the extra constant constraints result in that $\theta_1 = 0$.

(c) Argue why $g(x)$ is a cubic polynomial for $x \in [c_K, \infty)$ and that the extra constraints must result in that $g'(x) = 0$ in this interval.

(d) Show that the extra constraint results in the constraint

$$\sum_{k=1}^{K-2} \theta_{1+k}(c_{K-1} - c_k) = 0.$$

(e) Derive the basis functions in this case.

Exercise 9 (Multinomial regression)
Consider a setting where $Y \in \{0, 1, ..., K-1\}$ and we want to classify to one of the K categories based on some covariate vector $\boldsymbol{x}$. A possible model is then

$$\Pr(Y = k|\boldsymbol{x}) = \frac{\exp(\theta_{k,0} + \sum_{j=1}^{p} \theta_{k,j} x_j)}{\sum_{l=0}^{K-1} \exp(\theta_{l,0} + \sum_{j=1}^{p} \theta_{l,j} x_j)}, \quad k = 0, ..., K-1$$

(a) Show that the model above is equivalent to that

$$\Pr(Y = k|\boldsymbol{x}) = \frac{\exp(\beta_{k,0} + \sum_{j=1}^{p} \beta_{k,j} x_j)}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l,0} + \sum_{j=1}^{p} \beta_{l,j} x_j)}, \quad k = 1, ..., K-1$$

$$\Pr(Y = 0|\boldsymbol{x}) = 1 - \sum_{k=1}^{K-1} \Pr(Y = k|\boldsymbol{x})$$

for some relationships between the θ's and the β's. Discuss why it is better to consider the model based on the β's.

Assume now we have data $\{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)\}$ available and we want to estimate the β's

(b) Consider now only those observations for which $y_i \in \{0, k\}$. Define

$$z_i^k = \begin{cases} 1 & \text{if } y_i = k \\ 0 & \text{if } y_i = 0 \end{cases}$$

Based on the model above, show that (for $k > 0$)

$$\Pr(Z_i^k = 1|\boldsymbol{x}_i) = \frac{\exp(\beta_{k,0} + \sum_{j=1}^{p} \beta_{k,j} x_{ij})}{1 + \exp(\beta_{k,0} + \sum_{j=1}^{p} \beta_{k,j} x_{ij})}$$

Explain why we then are able to use logistic regression for estimation of the β parameters.

(c) For the `phoneme` dataset, divide the data into a training set and a test set

(i) Use logistic regression for estimating the β-parameters based on the training data. Calculate the error rate on the test data.

Hint: The `phoneme` dataset can be read into **R** by the command

```
ddir <- "http://www.uio.no/studier/emner/matnat/math/STK2100/v17/data/"
phoneme <- read.table(paste(ddir,"phoneme.data",sep=""),header=T,sep=",")
```

(ii) Compare the results obtained by the `multinom` routine.

Exercise 10 (Heart data)
In the textbook, the Heart data is used to produce figures 8.6 and 8.8. We will in this exercise see how these are constructed. The data are available from the course web-page under the file `heart.csv`

(*a*) Read the data into **R** and remove all observations where either the response or one of the covariates are not available.

(*b*) By modifying the commands in the `Carseats_tree.R` script, reproduce Figure 8.6 from the textbook.

(*c*) By using the option `ntree=B` in the `randomForrest` command, produce the black curve in Figure 8.8.

Why is the curve you obtain not exactly equal to the one in Figure 8.8?

(*d*) Now consider out-of-bag estimation error. Look at the help file for `predict.randomForest` to see how you can obtain the OOB estimates. Use this to produce the green curve in Figure 8.8. Plot this together with the curve from (*c*).

Exercise 11 (Rosenblatt's perceptron learning rule)
We will in this exercise look at a procedure, Rosenblatt's *perceptron learning rule* for constructing a separating hyperplane (if it exists). Let $\boldsymbol{X}$ be the design matrix, with one row for each individual observation and one column for each covariate (with the first column containing only 1's corresponding to the intercept). In addition we have a vector $\boldsymbol{y}$ where $y_i \in \{-1, 1\}$ for $i = 1, ..., n$. Our aim is to find a vector $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_p)$ such that

$$y_i \boldsymbol{\beta}^T \boldsymbol{x}_i > 0, \quad i = 1, ..., n. \tag{*}$$

(*a*) Assume that there exist a hyperplane that separates the two classes. Show that there then exist a vector $\boldsymbol{\beta}_{sep}$ such that

$$y_i \boldsymbol{\beta}_{sep}^T \boldsymbol{z}_i \geq 1 \tag{**}$$

where

$$\boldsymbol{z}_i = \boldsymbol{x}_i / ||\boldsymbol{x}_i||$$

for all i.

Hint: Show first that $\varepsilon = \min_i(y_i \boldsymbol{\beta}^T \boldsymbol{x}_i) > 0$.

(*b*) (This part is a bit hard.) Assume that we have a current value $\boldsymbol{\beta}$ (which do not separate the classes). Assume i is a point for which $y_i \boldsymbol{\beta}^T \boldsymbol{x}_i \leq 0$ and define

$$\boldsymbol{\beta}_{new} = \boldsymbol{\beta} + y_i \boldsymbol{z}_i$$

Show that

$$||\boldsymbol{\beta}_{new} - \boldsymbol{\beta}_{sep}||^2 \leq ||\boldsymbol{\beta} - \boldsymbol{\beta}_{sep}||^2 - 1$$

Hint: Use that $||y_i \boldsymbol{z}_i||^2 = 1$.

(*c*) Consider now the following algorithm:

(*i*) Choose an arbitratry $\boldsymbol{\beta}^0$ and define M^0 to be the set of i's for which $y_i \boldsymbol{\beta}^T \boldsymbol{x}_i < 0$ for $\boldsymbol{\beta} = \boldsymbol{\beta}^0$. Set $t = 0$

(*ii*) Continue until M^t is empty:

i. Select an arbitrary $i \in M^t$

ii. Set $\boldsymbol{\beta}^{t+1} = \boldsymbol{\beta}^t + y_i \boldsymbol{z}_i$

iii. Define M^{t+1} to be the set of i's for which $y_i \boldsymbol{\beta}^T \boldsymbol{x}_i < 0$ for $\boldsymbol{\beta} = \boldsymbol{\beta}^{t+1}$.

Based on the results in (*b*), show that this algorithm will converge in a fine number of iterations *if* a separating hyperplane exists.

(*d*) Implement a function having as input a matrix $\boldsymbol{X}$ of dimension $n \times p$ and a vector $\boldsymbol{y}$ of length n with $y_i \in \{-1, 1\}$ and which performs the algorithm above for finding a separating hyperplane.

(*e*) Consider the simulated data generated by

```
set.seed(1)
x=matrix(rnorm(20*2), ncol=2)
y=c(rep(-1,10), rep(1,10))
x[y==1,]=x[y==1,] + 2
```

Run the algorithm with different starting points and see how the algorithm performs. You can include a plot showing the different lines obtained at the various iterations of the algorithm.

(*f*) Modify the data so that there is no separating hyperplane. Try out the algorithm in this case and see what happens.

Exercise 12 (Neural networks on Spam data)
On the course web page the Spam data is available. Have a look on the `spam.info` file to get some information and background about these data.

(*a*) Read the data into **R** and use the binary variables in the `spam.traintest` to divide the data into a training and a test set (1 corresponding to training).

(*b*) Copy the commands using the `nnet` command from the `zip_nn.R` file in order to perform neural network classification using one hidden layer with 10 hidden variables. Test the resulting net on the test data and calculate the error rate.

Try out different values of the `decay` parameter. Also try out different number of hidden variables. What are the best choices you obtain?

(*c*) Now try out the `mlp` command again with one hidden layer and different number of hidden variables. Compare your results with the ones obtained by `nnet`.

If you get differences, try to explain why.

(*d*) Now extend to several layers of latent variables. Try out different options. For what combination do you get the best results?

(*e*) Now try to reduce the number of input variables by first performing logistic regression and model selection based on the AIC criterion. Use the remaining input variables within a neural network approach. What is your error rate in this case?

Hint: Use the `stepAIC` function.

(*f*) Summarize your results.

Exercise 13 (Neural network on `Wage` data)
We have earlier looked at the `Wage` data within the ISLR package. In this exercise we will see how we can use neural networks to predict wage.

(*a*) Start by making the data ready through the following commands. Note that the numerical covariates (input variables) are scaled to be between zero and one.

```
library(ISLR)
data(Wage)
set.seed(2)
n <- nrow(Wage)
train <- sample(1:n,n/2,replace=FALSE)
Wage$year = scale(Wage$year)
Wage$age = scale(Wage$age)
Wage.train <- Wage[train,]
Wage.test <- Wage[-train,]
```

(*b*) Now try out a neural network using the following commands:

```
m=10
wage.nnet = nnet(wage~year+age+maritl+race+education+jobclass+health+health_ins,
                 data=Wage.train,size=m,decay=0.1,linout=TRUE,
                 MaxNWts=10000,maxit=300)
pred.nnet = predict(wage.nnet,Wage.test)
resid = Wage.test$wage-pred.nnet
err.nnet = mean(resid^2)
```

Try out different values of of the decay parameter and number of latent variables.

(*c*) Now scale also the response variable by

```
#Scale response to be between zero and one.
maxWage = max(Wage$wage)
Wage$rwage = Wage$wage/maxWage
Wage.train <- Wage[train,]
Wage.test <- Wage[-train,]
```

Try out neural networks with `rwage` as response. Note that when doing predictions, you should multiply by `maxWage` in order to get it into the right scale again.

Compare the results with those obtained in (*b*). Discuss the differences.

(*d*) Turn now to deep learning. A problem in this case is that the `mpl` do not like factor variables as input variables. We can however transform these variables to dummy variables through the following commands:

```
X = model.matrix(~year+age+maritl+race+education+jobclass+health+health_ins,
                 data=Wage)
X.train = X[train,]
X.test = X[-train,]
```

We can then call `mpl` through the commands

```
library(RSNNS)
wage.dnet = mlp(X.train,Wage.train$rwage, size = c(10),linout=TRUE,
                learnFuncParams=c(0.3),maxit=300)
pred.dnet = maxWage*predict(wage.dnet,X.test)
resid = Wage.test$wage-pred.dnet
err.dnet = mean(resid^2)
```

Try these deep learning commands out with different networks. Try at least (10), $(10, 10)$, $(10, 10, 10)$ and $(10, 10, 10, 10)$.

(*e*) Consider now the best network from (*d*). Repeat the calls for this model. You should then see some variation. Try to explain why you get this variation.

(*f*) Given the variations in the mpl routine, a possibility is to fit several networks and then take the average of the predictions you obtain. Try this out using an average of 10 separate fitted networks. Report your results.

(*g*) Compare your results with previous prediction results on the `Wage` data.

Exercise 14

Consider a classification problem with $K = 3$ classes where Y denotes the class while the observations follow the distributions

$$X|Y = k \sim \text{Poisson}(\lambda_k)$$

Let $\lambda_1 = 10, \lambda_2 = 15$ and $\lambda_3 = 20$. Assume further that $\pi_k = \Pr(Y = k) = 1/K$ for all k.

(*a*) Derive the Bayes classifier in this case.

(*b*) Derive the error rate of the classifier.

Hint: First derive the error rate conditional on that $Y = k$.

(*c*) Write an **R** script that simulates (X, Y) 1000 times, calculates $\widehat{Y}$ based on the Bayes classifier.

Use this to estimate the error rate and compare with your result in (*b*).

Exercise 15
Assume a classification problem where $\Pr(Y = 1) = \Pr(Y = 2) = 0.5$ and

$$X|Y = k \sim N(\mu_k, 1)$$

with $\mu_1 = -1$ and $\mu_2 = 1$

(*a*) Derive the Bayes classifier in this case.

(*b*) Plot $\Pr(Y = 1|X = x)$ as a function of x. Discuss the behavior of this plot as $x \to \pm\infty$.

(*c*) Derive the *marginal* distribution $f_X(x)$ for X. Use this to construct a test where you reject

$$H_0 : X \sim f_X(x)$$

when X is extreme using a significance level α.

(*d*) Define now a classification rule where

$$\widehat{Y} = \begin{cases} \text{outlier} & \text{if } H_0 \text{ is rejected;} \\ \text{the Bayes classification} & \text{otherwise.} \end{cases}$$

Argue why the ordinary Bayes classifier corresponds to $\alpha = 0$.

Write an **R** function that performs such a classification.

(*e*) Simulates 1000 sets of (X, Y) from the specified distribution and perform the classification rule for different values of α (including $\alpha = 0$).

Discuss the results.

Exercise 16 (Dynamic and parallel computation in linear regression)
Consider first the simple model

$$y_i \overset{iid}{\sim} N(\mu, \sigma^2)$$

where the maximum likelihood estimates are given by

$$\begin{aligned} \hat{\mu} &= \frac{1}{n}\sum_{i=1}^{n} y_i \\ \hat{\sigma}^2 &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2 \end{aligned}$$

(*a*) Show that if we divide the data into groups where g_i indicates the group of observation i and n_g is the number of observations within group g, then

$$\hat{\mu} = \sum_{g=1}^{G} \frac{n_g}{n} \bar{y}_g \qquad \text{where } \bar{y}_g = \frac{1}{n_g} \sum_{i:g_i=g} y_i$$

and

$$\hat{\sigma}^2 = \sum_{g=1}^{G} \frac{n_g}{n} [\hat{\sigma}_g^2 + (\bar{y}_g - \bar{y})^2] \qquad \text{where } \hat{\sigma}_g^2 = \frac{1}{n_g} \sum_{i:g_i=g} (y_i - \bar{y}_g)^2$$

(*b*) Consider now dynamic calculation of the estimates. Denote the estimates based on the first s observations by $\hat{\mu}_s$ and $\hat{\sigma}_s^2$. Show that

$$\begin{aligned} \hat{\mu}_n &= \frac{1}{n} \sum_{i=1}^{n} y_i = \frac{n-1}{n} \hat{\mu}_{n-1} + \frac{1}{n} y_n \qquad (*) \\ \hat{\sigma}_n^2 &= \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})^2 = \frac{n-1}{n} \hat{\sigma}_{n-1}^2 + \frac{n-1}{n^2} (y_n - \bar{y}_{n-1})^2 \end{aligned}$$

Consider now the general linear regression model

$$y_i = \boldsymbol{x}_i^T \boldsymbol{\beta} + \varepsilon_i$$

where the least squares estimate is given by

$$\hat{\boldsymbol{\beta}} = [\boldsymbol{X}_n^T \boldsymbol{X}_n]^{-1} \boldsymbol{X}_n^T \boldsymbol{y}_n$$

where now $\boldsymbol{X}_s$ is the design matrix based on the first s observations and similarly $\boldsymbol{y}_s$ is the vector of the first s responses.

Define also $\boldsymbol{M}_n = \boldsymbol{X}_n^T \boldsymbol{X}_n$ and recall from Exercise 7 that

$$\boldsymbol{M}_n^{-1} = \boldsymbol{M}_{n-1}^{-1} - \frac{\tilde{\boldsymbol{x}}_n \boldsymbol{x}_n^T \boldsymbol{M}_{n-1}^{-1}}{1 + \boldsymbol{x}_n^T \tilde{\boldsymbol{x}}_n} \quad \text{where } \tilde{\boldsymbol{x}}_n = \boldsymbol{M}_{n-1}^{-1} \boldsymbol{x}_n$$

(*c*) Define now $\hat{\boldsymbol{\beta}}_s$ to be the least squares estimate based on the first s observations. Show that

$$\hat{\boldsymbol{\beta}}_n = [\boldsymbol{I} - \frac{\tilde{\boldsymbol{x}}_n \boldsymbol{x}_n^T}{1 + \boldsymbol{x}_n^T \tilde{\boldsymbol{x}}_n}][\hat{\boldsymbol{\beta}}_{n-1} + \tilde{\boldsymbol{x}}_n y_n] \qquad (**)$$

How much memory and how many operations are needed at each step in order to update $\hat{\boldsymbol{\beta}}_n$ in this way?

How would you initialize such an algorithm?

(*d*) Verify that (*) is a special case of (**).

Hint: Show first that $\boldsymbol{M}_n^{-1} = \frac{1}{n}$ in this case.

Exercise 17 (Adjustment for multiple testing)
Consider a linear regression model

$$Y = \beta_0 + \sum_{j=1}^{q} \beta_j x_j + \varepsilon$$

where q is large. We want to test

$$H_{0j} : \beta_j = 0 \text{ against } H_{0j} : \beta_j \neq 0$$

f or $j = 1, ..., q$.

(*a*) Assume all H_{0j}'s are true and $q = 1\,000\,000$. Performing a separate test for each j with significance level $\alpha = 0.01$, how many of the hypotheses will you expect to be rejected?

(*b*) One way of correcting for multiple testing is the Bonferroni correction. In this case each test is performed on significance level α/q instead.

Prove that this guarantees that if all H_{0j}'s are true, the probability of making at least one error is less or equal to α.

Discuss why this can lead to very low power (that is low probability for detecting a true H_{1j}).

Consider now the following table, which specifies the possible events that can happen where V is the number of true hypotheses that are rejected etc.

	H_{0j} true	H_{0j} wrong	Total
H_{0j} rejected	V	S	R
H_{0j} not rejected	U	T	q-R
Total	q_0	$q - q_0$	q

(*c*) Which quantities in the table above is stochastic?

Explain why the Bonferroni correction corresponds to controlling $\Pr(V > 0)$.

The FDR approach is based on another idea in that one wants to control the number of falsely rejected H_{0j} *among all hypotheses that are rejected*, that is $\frac{V}{R} = \frac{V}{V+S}$. In principle, one wants

$$E\left[\frac{V}{R}\right] = E\left[\frac{V}{V+S}\right] \leq \alpha \tag{1}$$

for any value of q_0 but this is a bit problematic directly, as we will see.

(*d*) Consider the case where $q_0 = q$. What are the values of $\frac{V}{V+S}$ that are possible in this case? Discuss why it is difficult to control $E\left[\frac{V}{R}\right]$ in this case.

(*e*) An alternative could be to control

$$E\left[\frac{V}{R}|R > 0\right]$$

Discuss why such a measure is also difficult to control.

The false discovery rate (FDR) approach is instead looking at

$$\Pr(R > 0)E\left[\frac{V}{R}\right]$$

The procedure for obtaining this is the following:

(*i*) Sort the p-values $p_1, ..., p_q$ corresponding to testing $H_{0j}, j = 1, ..., q$ to

$$p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(q)}$$

(*ii*) Find the largest k such that $p_{(k)} \leq \frac{k}{q}\alpha$

(*iii*) Reject $H_{0,j}$ if $p_j \leq p_{(k)}$

The proof on that this satisfies $\Pr(R > 0)E\left[\frac{V}{R}\right] \leq \alpha$ is somewhat technical, so we will rather study its properties through a simulation study.

(*f*) At the course web-page there is a routine `FDR.R` which contains commands for simulating data and performing testing based on the FDR approach.

Run these commands and plot the FDR as a function of q_0. Discuss the results.