# Compulsory Exercise 1 Solutions

TMA4268 Statistical Learning V2021

Emma Skarstein, Michail Spitieris, Stefanie Muff, Department of Mathematical Sciences, NTNU

Hand out date: February 8, 2021

---

**The submission deadline is Monday, February 22th 2021, 23:59h using Blackboard**

```
install.packages("knitr")  #probably already installed
install.packages("rmarkdown")  #probably already installed
install.packages("ggplot2")  #plotting with ggplot
install.packages("ggfortify")
install.packages("MASS")
install.packages("dplyr")
```

## Problem 1 (11P)

```
id <- "1X_8OKcoYbng1XvYFDirxjEWr7LtpNr1m"  # google file ID
values <- dget(sprintf("https://docs.google.com/uc?id=%s&export=download", id))

X = values$X
dim(X)
```

```
## [1] 100  81
```

```
x0 = values$x0
dim(x0)
```

```
## [1] 81  1
```

```
beta = values$beta
dim(beta)
```

```
## [1] 81  1
```

```
sigma = values$sigma
sigma
```

```
## [1] 0.5
```

**Solution**

### a)

$$E(\widetilde{\boldsymbol{\beta}}) = E[(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}] = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T E(\mathbf{Y}) = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}$$

$$\text{Var}(\widetilde{\boldsymbol{\beta}}) = \text{Var}[(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}] = (\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\text{Var}(\mathbf{Y})\mathbf{X}(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1} = \sigma^2(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}$$

## b)

$$E(\mathbf{x}_0^T\widetilde{\boldsymbol{\beta}}) = \mathbf{x}_0^T(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}$$

and

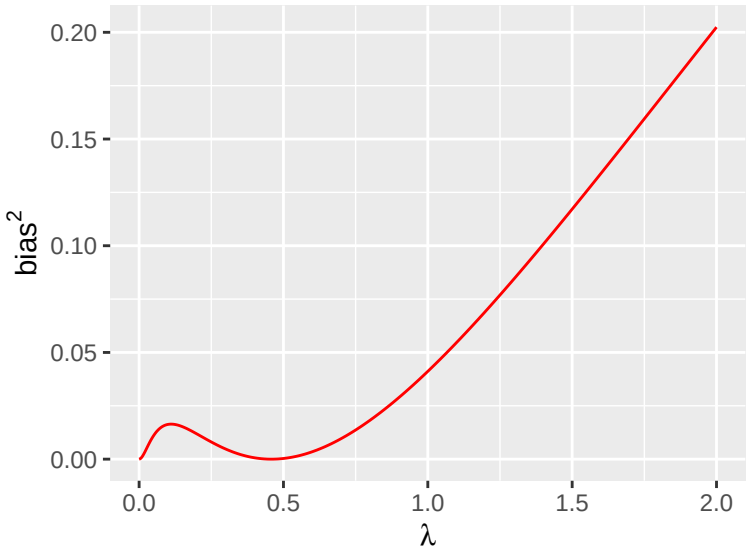
$$\text{Var}(\mathbf{x}_0^T\widetilde{\boldsymbol{\beta}}) = \sigma^2\mathbf{x}_0^T(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{x}_0$$

## c)

$$\text{E}[(Y_0 - \widetilde{f}(\mathbf{x}_0))^2] = (\mathbf{x}_0^T\boldsymbol{\beta} - \mathbf{x}_0^T(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}\boldsymbol{\beta})^2 + \mathbf{x}_0^T\sigma^2(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{x}_0 + \sigma^2$$

## d)

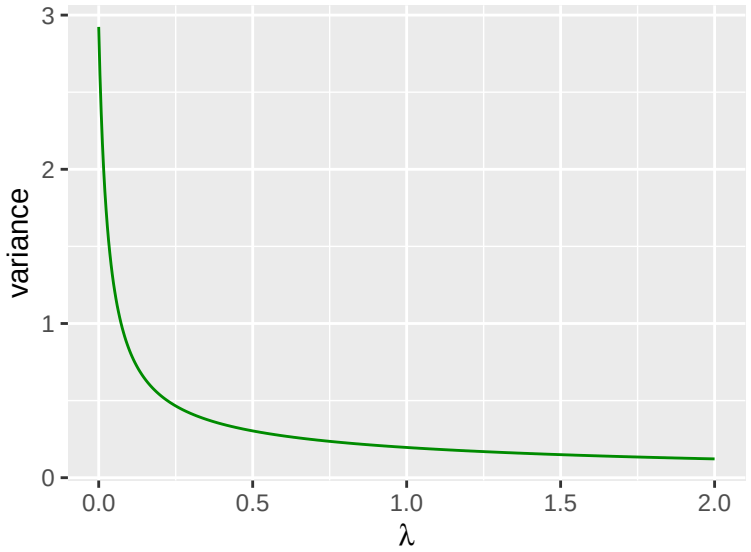
```
library(ggplot2)
bias = function(lambda, X, x0, beta) {
    p = ncol(X)
    value = (t(x0) %*% solve(t(X) %*% X + lambda * diag(p)) %*% t(X) %*% X %*% beta -
        t(x0) %*% beta)^2
    return(value)
}
lambdas = seq(0, 2, length.out = 500)
BIAS = rep(NA, length(lambdas))
for (i in 1:length(lambdas)) BIAS[i] = bias(lambdas[i], X, x0, beta)
dfBias = data.frame(lambdas = lambdas, bias = BIAS)
ggplot(dfBias, aes(x = lambdas, y = bias)) + geom_line(color = "red") + xlab(expression(lambda)) +
    ylab(expression(bias^2))
```




This is expected, since as $\lambda$ increase we introduce more bias in the etsimator. Mybe we didn't expect this bumb in the start.

## e)

```
variance = function(lambda, X, x0, sigma) {
    p = ncol(X)
    inv = solve(t(X) %*% X + lambda * diag(p))
    value = sigma^2 * (t(x0) %*% inv %*% t(X) %*% X %*% inv %*% x0)
    return(value)
}
lambdas = seq(0, 2, length.out = 500)
VAR = rep(NA, length(lambdas))
for (i in 1:length(lambdas)) VAR[i] = variance(lambdas[i], X, x0, sigma)
dfVar = data.frame(lambdas = lambdas, var = VAR)
ggplot(dfVar, aes(x = lambdas, y = var)) + geom_line(color = "green4") + xlab(expression(lambda)) +
    ylab("variance")
```



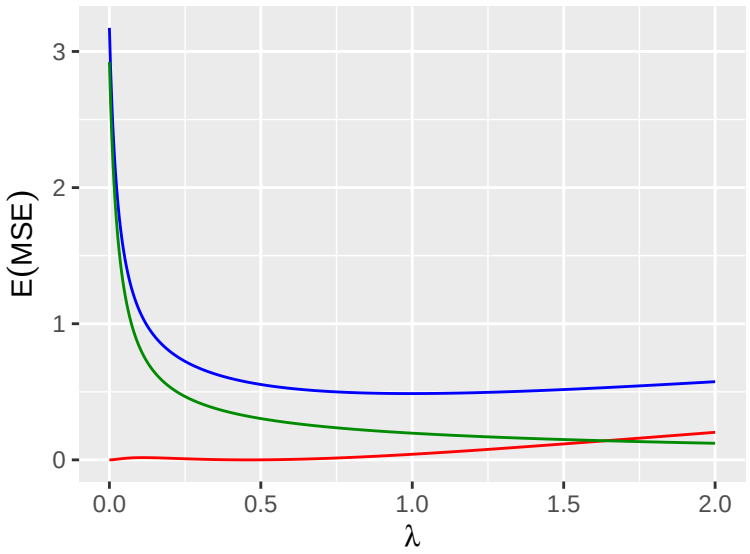

This is also expected, since with increasing $\lambda$ the data are given less weight.

## f)

```
exp_mse = BIAS + VAR + sigma^2
lambdas[which.min(exp_mse)]
```

```
## [1] 0.993988
```

```
dfAll = data.frame(lambda = lambdas, bias = BIAS, var = VAR, exp_mse = exp_mse)
ggplot(dfAll) + geom_line(aes(x = lambda, y = exp_mse), color = "blue") + geom_line(aes(x = lambda,
    y = bias), color = "red") + geom_line(aes(x = lambda, y = var), color = "green4") +
    xlab(expression(lambda)) + ylab(expression(E(MSE)))
```

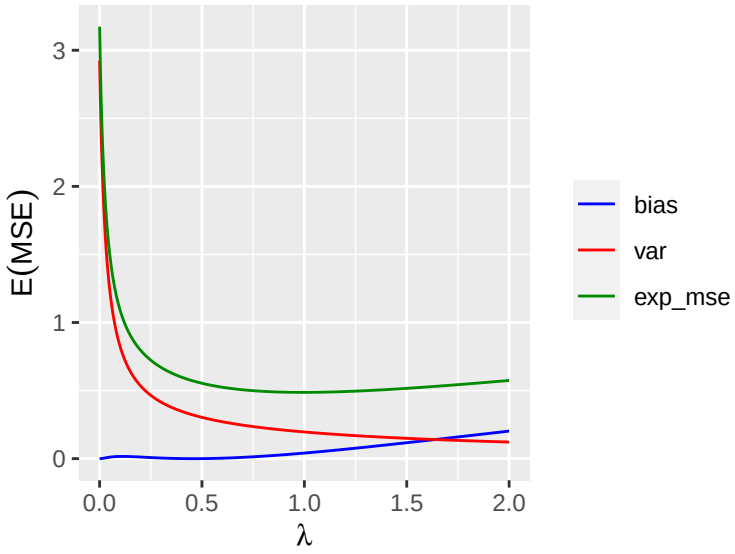



Plot for those who want to add a label:

```
dfAll = data.frame(lambda = lambdas, bias = BIAS, var = VAR, exp_mse = exp_mse)
dfAll_melt = reshape2::melt(data = dfAll, id.vars = "lambda")
names(dfAll_melt)
```

```
## [1] "lambda"   "variable" "value"
```

```
ggplot(dfAll_melt, aes(x = lambda, y = value, color = variable)) + geom_line() +
    xlab(expression(lambda)) + ylab(expression(E(MSE))) + theme(legend.title = element_blank()) +
    scale_color_manual(values = c("blue", "red", "green4"))
```




# Problem 2 - 13P

## a)

```r
id <- "1yYlEl5gYY3BEtJ4d7KWaFGIOEweJIn__"  # google file ID
d.corona <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    id), header = T)
```

```r
table(d.corona$deceased)
```

```
##
##    0    1
## 1905  105
```

```r
table(d.corona$country, d.corona$sex)
```

```
##
##             female male
##   France        60   54
##   indonesia     30   39
##   japan        120  174
##   Korea        879  654
```

```r
table(d.corona$deceased, d.corona$sex)
```

```
##
##     female male
##   0   1046  859
##   1     43   62
```

```r
France = d.corona[which(d.corona$country == "France"), ]

table(France$deceased, France$sex)
```

```
##
##     female male
##   0     55   43
##   1      5   11
```

## b)

1P point for choosing the right regression model, and 1P each for corretly answering (i) – (iv).

The response (deceased) is binary, so we need a logistic regression model with sex, age and country in the model together (no point given when they fit models with only one single variable!)

```r
# Logistic regression
r.glm <- glm(deceased ~ sex + age + country, d.corona, family = "binomial")
summary(r.glm)
```

```
##
## Call:
## glm(formula = deceased ~ sex + age + country, family = "binomial",
##     data = d.corona)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9050  -0.3508  -0.2761  -0.2144   3.1165
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)      -3.993485   0.462190  -8.640  < 2e-16 ***
## sexmale           0.626068   0.209045   2.995  0.00275 **
## age               0.027134   0.004736   5.729 1.01e-08 ***
## countryindonesia -0.411855   0.550051  -0.749  0.45400
## countryjapan     -1.343383   0.417196  -3.220  0.00128 **
## countryKorea     -0.773895   0.307980  -2.513  0.01198 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 824.32  on 2009  degrees of freedom
## Residual deviance: 766.16  on 2004  degrees of freedom
## AIC: 778.16
##
## Number of Fisher Scoring iterations: 6
```

```
anova(r.glm, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: deceased
##
## Terms added sequentially (first to last)
##
##
##         Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                     2009     824.32
## sex      1    7.785      2008     816.54  0.005268 **
## age      1   39.311      2007     777.23 3.613e-10 ***
## country  3   11.063      2004     766.16  0.011390 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(i) For the probability to die for certain set of covariates, we have to use that $p = \exp(\eta)/(1 + \exp \eta)$, that is

```
coefs <- summary(r.glm)$coef[, 1]

eta <- coefs["(Intercept)"] + coefs["sexmale"] + 75 * coefs["age"] + coefs["countryKorea"]

(prob <- exp(eta)/(1 + exp(eta)))
```

```
## (Intercept)
##   0.1084912
```

(ii) Yes, there is quite strong evidence that males have higher probability to die. The coefficient is positive and $p = 0.00275$ (the full point is only given if they give an explanation).

(iii) Yes. Important that we need the anova table here! Point is only given if the anova table is presented and mentioned that $p = 0.0114$ gives evidence that the country is relevant. No point is given if the anova table is missing!

(iv) The odds ratio between two otherwise identical individuals with an age difference of 10 years can be calculated as:

```r
exp(coefs["age"] * 10)
```

```
##      age
## 1.311724
```

The correct answer is therefore that the odds (not the odds ratio!) to die increased by a factor of 1.31.

## c)

In both (i) and (ii) the models should contain all three covariates (sex, age, country), plus the respective interaction terms. Students should NOT remove any variables, because all three are relevant for correct inference. Thus if e.g. in (i) they don't include country, -1P has to be deducted.

(i) The correct model contains all three covariates and an interaction between sex and age

```r
mod2 = glm(deceased ~ sex * age + country, d.corona, family = "binomial")
summary(mod2)
```

```
##
## Call:
## glm(formula = deceased ~ sex * age + country, family = "binomial",
##     data = d.corona)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9111  -0.3500  -0.2768  -0.2150   3.1078
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -3.953580   0.571712  -6.915 4.67e-12 ***
## sexmale           0.556745   0.624834   0.891 0.372913
## age               0.026485   0.007261   3.648 0.000265 ***
## countryindonesia -0.410197   0.550304  -0.745 0.456030
## countryjapan     -1.344440   0.417364  -3.221 0.001276 **
## countryKorea     -0.772596   0.308244  -2.506 0.012195 *
## sexmale:age       0.001111   0.009443   0.118 0.906350
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 824.32  on 2009  degrees of freedom
## Residual deviance: 766.15  on 2003  degrees of freedom
## AIC: 780.15
##
## Number of Fisher Scoring iterations: 6
```

The answer is thus **no**. The slope for women is $\beta_2$, while the slope for males is $\beta_2 + \beta_6$. As we have no evidence that $\beta_6$ is different from zero ($p = 0.906$), we have no evidence that age is a greater risk for males than for females.

(ii) The correct model contains all three covariates and an interaction between age and country:

```r
mod3 = glm(deceased ~ sex + age * country, d.corona, family = "binomial")
summary(mod3)
```

```
##
## Call:
## glm(formula = deceased ~ sex + age * country, family = "binomial",
##     data = d.corona)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2681  -0.3447  -0.2768  -0.2180   3.0170
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -7.04272    1.72789  -4.076 4.58e-05 ***
## sexmale               0.62777    0.21056   2.981  0.00287 **
## age                   0.06693    0.02124   3.151  0.00163 **
## countryindonesia      4.34249    2.16594   2.005  0.04497 *
## countryjapan          2.13091    2.03299   1.048  0.29456
## countryKorea          2.37162    1.75357   1.352  0.17623
## age:countryindonesia -0.07189    0.03310  -2.172  0.02986 *
## age:countryjapan     -0.04630    0.02668  -1.736  0.08260 .
## age:countryKorea     -0.04142    0.02189  -1.892  0.05854 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 824.32  on 2009  degrees of freedom
## Residual deviance: 759.71  on 2001  degrees of freedom
## AIC: 777.71
##
## Number of Fisher Scoring iterations: 6
```

The answer is thus **yes**. The slope for age for the Indonesian population is $\beta_2 + \beta_6$, while it is $\beta_2$ for French. Given that $p = 0.030$ for $\beta_6$, there is evidence that age is a smaller risk factor in Indonesia – thus a larger risk factor in France.

## d)

TRUE – TRUE – TRUE – FALSE

```
library(MASS)
sum(d.corona$deceased)/nrow(d.corona)
```

```
## [1] 0.05223881
```

```
table(predict = predict(lda(deceased ~ age + sex + country, data = d.corona))$class,
    true = d.corona$deceased)
```

```
##        true
## predict    0    1
##       0 1905  105
##       1    0    0
```

```
table(predict = predict(qda(deceased ~ age + sex + country, data = d.corona))$class,
    true = d.corona$deceased)
```

```
##        true
```

```
## predict    0    1
##       0 1751   84
##       1  154   21
```

LDA has a specificity of 1, but a sensitivity of 0, so it is useless. QDA has a somewhat higher sensitivity.

# Problem 3 - 15P

In this problem, we will use a data set regarding the presence of *diabetes* from a population of women of Pima heritage in the US. For each woman we have we have the following information:

*diabetes*: 1 = present, 0 = not present

*npreg*: number of pregnacies

*glu*: plasma glucose concentration in an oral glucose tolerance test

*bp*: diastolic blood pressure (mm Hg)

*skin*: triceps skin fold thickness (mm)

*bmi*: body mass index

*ped*: diabetes pedigree function

*age*: age in years

The aim is to use the methods you have learned so far in order to make a classication rule for diabetes (or not) based on the available data. You will use the training set, `train` to fit the models and at the end you will use the test set, `test` to compare the fitted models. The training set consists of 300 observations, 200 non-diabetes and 100 diabetes cases and the testing set includes 232 observations, 155 non-diabetes and 77 diabetes cases.

```
# read file
diab = dget("https://www.math.ntnu.no/emner/TMA4268/2019v/data/flying.dd")
t = MASS::Pima.tr2
train = diab$ctrain
test = diab$ctest
```

We will first create a logistic regression model where the probability to win for player 1 has the form

$$P(Y_i = 1|\mathbf{X} = \boldsymbol{x}_i) = p_i = \frac{e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\beta_3 x_{i3}+\beta_4 x_{i4}}}{1 + e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\beta_3 x_{i3}+\beta_4 x_{i4}}} ,$$

where $x_{i1}$ is the number of aces for player 1 in match $i$, $x_{i2}$ is the number of aces for player 2 in match $i$, and $x_{i3}$ and $x_{i4}$ are the number of unforced errors commited by player 1 and 2 in match $i$. $Y_i = 0$ represents player 1 winning match $i$, $Y_i = 0$ represents player 1 loosing match $i$.

## a)

```
# read file
diab = dget("https://www.math.ntnu.no/emner/TMA4268/2019v/data/flying.dd")
t = MASS::Pima.tr2
train = diab$ctrain
test = diab$ctest
```

We first fit a logistic regression model where the probability of diabetes is given by

$$P(y_i = 1|\mathbf{X} = \mathbf{x}_i) = p_i = \frac{e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_7 x_{i7}}}{1 + e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_7 x_{i7}}},$$

where $y_i = 1$ is presence of diabetes ($y_i = 0$ is non-presence), $x_{i1}$ is the value of *npreg* for the $i$th observation, $x_{i2}$ is the value of *glu*, et cetera.

```
logReg = glm(diabetes ~ ., data = train, family = "binomial")
summary(logReg)
```

```
##
## Call:
## glm(formula = diabetes ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8155  -0.6367  -0.3211   0.6147   2.2408
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.583538   1.428276  -7.410 1.26e-13 ***
## npreg         0.105109   0.062721   1.676 0.093775 .
## glu           0.035586   0.005892   6.039 1.55e-09 ***
## bp           -0.014654   0.013982  -1.048 0.294615
## skin          0.020379   0.020575   0.990 0.321962
## bmi           0.094683   0.031265   3.028 0.002458 **
## ped           1.931666   0.529573   3.648 0.000265 ***
## age           0.038291   0.020247   1.891 0.058594 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 381.91  on 299  degrees of freedom
## Residual deviance: 253.84  on 292  degrees of freedom
## AIC: 269.84
##
## Number of Fisher Scoring iterations: 5
```

(i) First we write the linear predictor as $\eta_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_7 x_{i7}$. Then, we can write that $p_i = \frac{e^{\eta_i}}{1+e^{\eta_i}}$ and we have that

$$\begin{aligned} p_i + p_i e^{\eta_i} &= e^{\eta_i} \\ e^{\eta_i}(1 - p_i) &= p_i \\ \frac{p_i}{1 - p_i} &= e^{\eta_i} \\ \log(\frac{p_i}{1 - p_i}) &= \eta_i \end{aligned} \tag{1}$$

(ii) 0.5P for the correct predictions, 0.5 for the confusion table, 0.5P for sensitivity an 0.5P for specificity.

```
predLogistic = predict(logReg, newdata = test, type = "response")
logClass = round(predLogistic)
ta = table(test$diabetes, logClass)
ta
```

```
##    logClass
```

```
##       0   1
##   0 137  18
##   1  29  48
```

```
sensLog = ta[2, 2]/(sum(ta[2, ]))
spesLog = ta[1, 1]/sum(ta[1, ])
c(sensitivity = sensLog, specificity = spesLog)
```

```
## sensitivity specificity
##   0.6233766   0.8838710
```

## b)

(i) 0.5P for each. $\pi_k$ is the prior probability that a randomly chosen observation is a diabetes case ($\pi_1$) or not ($\pi_0$). We assume that the observation in the diabetes (1) and non-diabetes (0) cases comes from a multivariate normal distribution $f_k(\mathbf{x}) \sim \mathsf{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}_1$ is the mean vector of the diabetes cases and $\boldsymbol{\mu}_0$ is the mean vector of the non-cases. The variance-covariance matrix $\boldsymbol{\Sigma}$ is the same in both groups.

(ii) 1P for fitting LDA and QDA (0.5P each), 1P for the confusion tables (0.5P each), and 1P for the explanation between LDA and QDA.

The difference between LDA and QDA is that the first assumes a common covariance matrix for all classes in the data while the later assumes that each class has its own covariance matrix. This leads to a linear decision boundary in LDA and a quadratic decision boundary in QDA. QDA is thus more flexible than LDA, but also requires that more parameters are estimated.

### LDA

```
library(MASS)
ldaMod = lda(diabetes ~ ., data = train)
postLDA = predict(ldaMod, newdata = test)$posterior  # posterior probabilities
predLDA = predict(ldaMod, newdata = test)$class  # predicted class | round(postLDA)
tLDA = table(test$diabetes, predLDA)
tLDA
```

```
##    predLDA
##       0   1
##   0 138  17
##   1  30  47
```

```
sensLDA = tLDA[2, 2]/(sum(tLDA[2, ]))
spesLDA = tLDA[1, 1]/(sum(tLDA[1, ]))
c(sensitivity = sensLDA, specificity = spesLDA)
```

```
## sensitivity specificity
##   0.6103896   0.8903226
```

### QDA

```
library(MASS)
qdaMod = qda(diabetes ~ ., data = train)
postQDA = predict(qdaMod, newdata = test)$posterior
predQDA = predict(qdaMod, newdata = test)$class
tQDA = table(test$diabetes, predQDA)
tQDA
```

```
##    predQDA
##       0   1
##   0 131  24
##   1  32  45
```

```
sensQDA = tQDA[2, 2]/(sum(tQDA[2, ]))
spesQDA = tQDA[1, 1]/(sum(tQDA[1, ]))
c(sensitivity = sensQDA, specificity = spesQDA)
```

```
## sensitivity specificity
##   0.5844156   0.8451613
```

## c)

(i) First, we compute the Euclidean distance between the new observation and all the observations in the data. Then we identify the $k$ nearest neigbours and we classify the new observation according to the class "majority vote" of the $k$ neighbours.

(ii) The tuning parameter $k$ can be decided by cross validation.

(iii) 0.5P or the knn code, 0.5P for the confusion table, 0.5P for sensitivity and 0.5P for specificity.

```
library(class)
trainKNN = subset(train, select = -diabetes)
testKNN = subset(test, select = -diabetes)

knnMod = knn(train = trainKNN, test = testKNN, cl = train$diabetes, k = 25, prob = T)

(tKNN = table(true = test$diabetes, predicted = knnMod))
```

```
##     predicted
## true   0   1
##    0 144  11
##    1  36  41
```

```
sensKNN = tKNN[2, 2]/(sum(tKNN[2, ]))
spesKNN = tKNN[1, 1]/(sum(tKNN[1, ]))
c(sensitivity = sensKNN, specificity = spesKNN)
```

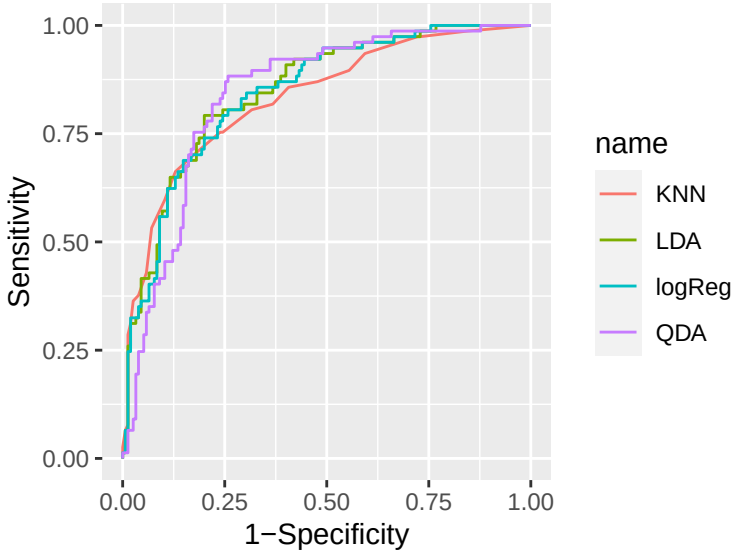
```
## sensitivity specificity
##   0.5324675   0.9290323
```

## d)

```
library(pROC)
library(plotROC)

logReg.ROC = roc(response = test$diabetes, predictor = predLogistic)

LDA.ROC = roc(response = test$diabetes, predictor = postLDA[, 2])
QDA.ROC = roc(response = test$diabetes, predictor = postQDA[, 2])

# The probabilities in the knnMod output are for the respective categorized
# class, and not directly P(y=1), so we need to use 1-prob if the class was
# categorized as 0. We generate the probKNN vector with the respective correct
```

```r
# information that we need for the ROC curve:
probKNN = ifelse(knnMod == 0, 1 - attributes(knnMod)$prob, attributes(knnMod)$prob)
KNN.ROC = roc(response = test$diabetes, predictor = probKNN)
```

```r
probs = data.frame(diabetes = test$diabetes, logReg = predLogistic, LDA = postLDA[,
    2], QDA = postQDA[, 2], KNN = probKNN)
plProbs = melt_roc(probs, "diabetes", c("logReg", "LDA", "QDA", "KNN"))
ggplot(plProbs, aes(d = D, m = M, color = name)) + geom_roc(n.cuts = F, size = 0.5) +
    xlab("1-Specificity") + ylab("Sensitivity")
```




```r
aucAll = data.frame(auc = c(auc(logReg.ROC), auc(LDA.ROC), auc(QDA.ROC), auc(KNN.ROC)))
rownames(aucAll) = c("logReg", "LDA ", "QDA ", "KNN ")
kableExtra::kable(aucAll)
```

| | auc |
|---|---|
| logReg | 0.8450775 |
| LDA | 0.8490155 |
| QDA | 0.8414747 |
| KNN | 0.8325513 |

According to AUC, all four methods perform similarly well. Although LDA performs better, the difference in the performance with respect to logistic regression is negligible. If the aim was to create an interprentable model, we would thus choose logistic regression.

# Problem 4 (6P)

# Answer

## a)

The LOOCV statistic is defined as $\text{CV} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_{(-i)})^2$ where $\hat{y}_{(-i)} = \mathbf{x}_i^T\hat{\boldsymbol{\beta}}_{(-i)}$

The equation for $\hat{\boldsymbol{\beta}}_{(-i)}$ is the same as for $\hat{\boldsymbol{\beta}}$ but by deleting the $i$th row of the $X$ matrix and is written as

$$\hat{\boldsymbol{\beta}}_{(-i)} = (X_{(-i)}^T X_{(-i)})^{-1} X_{(-i)}^T \mathbf{y}_{(-i)}.$$

From Hint 2 we have that $X_{(-i)}^T X_{(-i)} = X^T X - \mathbf{x}_i \mathbf{x}_i^T$, and by using the Sherman–Morrison formula (Hint 3) we have that

$$\begin{aligned}
(X_{(-i)}^T X_{(-i)})^{-1} &= (X^T X - \mathbf{x}_i \mathbf{x}_i^T)^{-1} \\
&= (X^T X)^{-1} + \frac{(X^T X)^{-1} \mathbf{x}_i \mathbf{x}_i^T (X^T X)^{-1}}{1 - \mathbf{x}_i^T (X^T X)^{-1} \mathbf{x}_i} \\
&= (X^T X)^{-1} + \frac{(X^T X)^{-1} \mathbf{x}_i \mathbf{x}_i^T (X^T X)^{-1}}{1 - h_i}
\end{aligned} \tag{2}$$

Using Hint 2.1 we have that $X_{(-i)}^T \mathbf{y}_{(-i)} = X^T \mathbf{y} - \mathbf{x}_i y_i$ and

$$\begin{aligned}
\hat{\boldsymbol{\beta}}_{(-i)} &= (X_{(-i)}^T X_{(-i)})^{-1} X_{(-i)}^T \mathbf{y}_{(-i)} \\
&= \left( (X^T X)^{-1} + \frac{(X^T X)^{-1} \mathbf{x}_i \mathbf{x}_i^T (X^T X)^{-1}}{1 - h_i} \right) \left( X^T \mathbf{y} - \mathbf{x}_i y_i \right) \\
&= \hat{\boldsymbol{\beta}} - (X^T X)^{-1} \mathbf{x}_i y_i + \frac{(X^T X)^{-1} \mathbf{x}_i \mathbf{x}_i^T \hat{\boldsymbol{\beta}} - (X^T X)^{-1} \mathbf{x}_i h_i y_i}{1 - h_i} \\
&= \hat{\boldsymbol{\beta}} - \frac{(X^T X)^{-1} \mathbf{x}_i}{1 - h_i} [y_i (1 - h_i) - \hat{y}_i + h_i y_i] \\
&= \hat{\boldsymbol{\beta}} - \frac{(X^T X)^{-1} \mathbf{x}_i (y_i - \hat{y}_i)}{1 - h_i}
\end{aligned} \tag{3}$$

Finally, by substituting $\hat{\boldsymbol{\beta}}_{(-i)}$ in the equation $\hat{y}_{(-i)} = \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_{(-i)}$ and we have that

$$\begin{aligned}
\hat{y}_{(-i)} &= \mathbf{x}_i^T \hat{\boldsymbol{\beta}} - \frac{\mathbf{x}_i^T (X^T X)^{-1} \mathbf{x}_i (y_i - \hat{y}_i)}{1 - h_i} \\
&= \hat{y}_i - \frac{h_i (y_i - \hat{y}_i)}{1 - h_i}
\end{aligned} \tag{4}$$

and

$$\begin{aligned}
y_i - \hat{y}_{(-i)} &= y_i - \hat{y}_i + \frac{h_i (y_i - \hat{y}_i)}{1 - h_i} \\
&= (y_i - \hat{y}_i)(1 + \frac{h_i}{1 - h_i}) \\
&= \frac{y_i - \hat{y}_i}{1 - h_i}
\end{aligned} \tag{5}$$

$\text{CV} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_{(-i)})^2 = \frac{1}{N} \left( \frac{y_i - \hat{y}_i}{1 - h_i} \right)^2$.

## b)

(i) F
(ii) T
(iii) F
(iv) F

## Problem 5 (5P)

```
id <- "19auu8YlUJJJUsZY8JZfsCTWzDm6doE7C"  # google file ID
d.bodyfat <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    id), header = T)
```

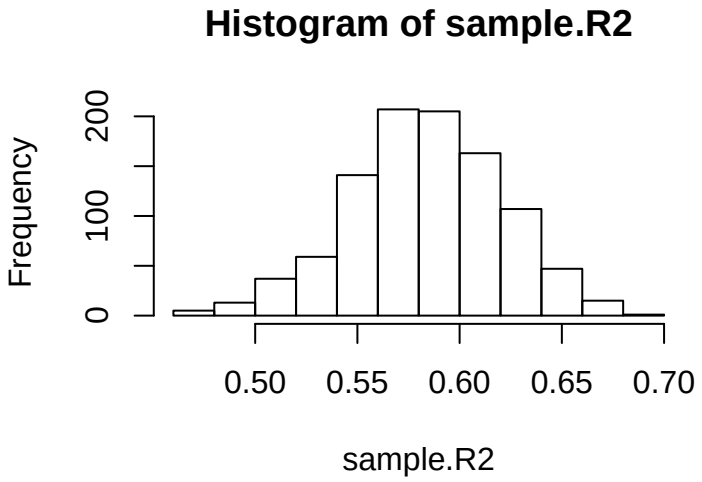
### a) (1P)

```
r.lm <- lm(bodyfat ~ age + weight + bmi, data = d.bodyfat)
summary(r.lm)$r.squared
```

```
## [1] 0.5803041
```

### b) (4P)

```
set.seed(4268)
B <- 1000
sample.R2 <- rep(NA, B)
for (ii in 1:B) {
    d.boot <- d.bodyfat[sample(1:nrow(d.bodyfat), size = nrow(d.bodyfat), replace = TRUE),
        ]
    r.lm.boot <- lm(bodyfat ~ age + weight + bmi, data = d.boot)
    sample.R2[ii] <- summary(r.lm.boot)$r.squared
}
hist(sample.R2)
```




The standard error and 95% CI are just sample SE and the respective quantiles:

```
sd(sample.R2)
```

```
## [1] 0.03705002
```

```
quantile(sample.R2, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.5090717 0.6534133
```

Interpretation: There is uncertainty in $R^2$, although that is usually not visible/reported in the summary output.