Compulsory exercise 1: Group 4

TMA4268 Statistical Learning V2021

Øystein Alvestad, Lars Evje and William Scott Grundeland Olsen

24 februar, 2021

Problem 1

We consider the regression problem

$$Y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon,$$

where $\mathrm{E}(\varepsilon) = 0$ and $\mathrm{Var}(\varepsilon) = \sigma^2$. We define $\mathbf{x}, \boldsymbol{\beta} \in \mathbb{R}^{p+1}$ such that $f(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\beta} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$.

a)

We consider the estimator

$$\tilde{\boldsymbol{\beta}} = (X^\top X + \lambda I)^{-1} X^\top \mathbf{Y},$$

for $\boldsymbol{\beta}$. Here X is the design matrix, $\mathbf{Y}$ is the response vector, I is the identity matrix, and $\lambda \geq 0$ is a constant. We recall that for a constant matrix A of appropriate dimensions, and a random vector $\mathbf{Z}$, we have that

$$\mathrm{E}(A\mathbf{Z}) = A\,\mathrm{E}(\mathbf{Z}) \quad \text{and} \quad \mathrm{Var}(A\mathbf{Z}) = A\,\mathrm{E}(\mathbf{Z})A^\top. \tag{1}$$

First we now find the expected value

$$\mathrm{E}(\tilde{\boldsymbol{\beta}}) = \mathrm{E}((X^\top X + \lambda I)^{-1} X^\top \mathbf{Y}) = (X^\top X + \lambda I)^{-1} X^\top\,\mathrm{E}(\mathbf{Y}),$$

and because $\mathrm{E}(\mathbf{Y}) = X\boldsymbol{\beta}$, we get that

$$\mathrm{E}(\tilde{\boldsymbol{\beta}}) = (X^\top X + \lambda I)^{-1} X^\top X \boldsymbol{\beta}.$$

Similarly, the variance-covariance matrix is

$$\mathrm{Var}(\tilde{\boldsymbol{\beta}}) = \mathrm{Var}((X^\top X + \lambda I)^{-1} X^\top \mathbf{Y}) = (X^\top X + \lambda I)^{-1} X^\top\,\mathrm{Var}(\mathbf{Y})[(X^\top X + \lambda I)^{-1} X^\top]^\top,$$

and because $\mathrm{Var}(\mathbf{Y}) = \sigma^2 I$, we get that

$$\mathrm{Var}(\tilde{\boldsymbol{\beta}}) = \sigma^2 (X^\top X + \lambda I)^{-1} X^\top X (X^\top X + \lambda I)^{-1},$$

where we used $(B^{-1})^\top = (B^\top)^{-1}$, for an invertible matrix B.

b)

We now let $\tilde{f}(\mathbf{x}_0) = \mathbf{x}_0^\top \tilde{\boldsymbol{\beta}}$ be the prediction at a new covariate vector $\mathbf{x}_0$, and we wish the expected value and variance of this. Using what we learned in **a)** and Equation (1), we find that

$$\mathrm{E}(\tilde{f}(\mathbf{x}_0)) = \mathrm{E}(\mathbf{x}_0^\top \tilde{\boldsymbol{\beta}}) = \mathbf{x}_0^\top\,\mathrm{E}(\tilde{\boldsymbol{\beta}}) = \mathbf{x}_0^\top (X^\top X + \lambda I)^{-1} X^\top X \boldsymbol{\beta}.$$

Similarly, we find that

$$\mathrm{Var}(\tilde{f}(\mathbf{x}_0)) = \mathrm{Var}(\mathbf{x}_0^\top \tilde{\boldsymbol{\beta}}) = \mathbf{x}_0^\top\,\mathrm{Var}(\tilde{\boldsymbol{\beta}})\mathbf{x}_0 = \sigma^2 \mathbf{x}_0^\top (X^\top X + \lambda I)^{-1} X^\top X (X^\top X + \lambda I)^{-1} \mathbf{x}_0. \tag{2}$$

c)

We want to find the expected MSE at $\mathbf{x}_0$, and this is most easily done using the relationship between the expected value and the variance-covariance matrix,

$$\mathrm{E}((y_0 - \tilde{f}(\mathbf{x}_0))^2) = \mathrm{E}(\tilde{f}(\mathbf{x}_0) - f(\mathbf{x}_0))^2 + \mathrm{Var}(\tilde{f}(\mathbf{x}_0)) + \mathrm{Var}(\varepsilon).$$

The last two terms are known to us now, so we look further at the first term, the squared bias,

$$\begin{aligned} \mathrm{E}(\tilde{f}(\mathbf{x}_0) - f(\mathbf{x}_0))^2 &= [\mathrm{E}(\tilde{f}(\mathbf{x}_0)) - \mathrm{E}(f(\mathbf{x}_0))]^2 = [\mathrm{E}(\tilde{f}(\mathbf{x}_0)) - f(\mathbf{x}_0)]^2 \\ &= [\mathbf{x}_0^\top (X^\top X + \lambda I)^{-1} X^\top X \boldsymbol{\beta} - \mathbf{x}_0^\top \boldsymbol{\beta}]^2. \end{aligned} \tag{3}$$

We then get that

$$\begin{aligned} \mathrm{E}((y_0 - \tilde{f}(\mathbf{x}_0))^2) = \sigma^2 &+ [\mathbf{x}_0^\top (X^\top X + \lambda I)^{-1} X^\top X \boldsymbol{\beta} - \mathbf{x}_0^\top \boldsymbol{\beta}]^2 \\ &+ \sigma^2 \mathbf{x}_0^\top (X^\top X + \lambda I)^{-1} X^\top X (X^\top X + \lambda I)^{-1} \mathbf{x}_0. \end{aligned}$$

d)

We start by importing the relevant quantities, as given in the project description.

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

X <- values$X
x0 <- values$x0
beta <- values$beta
sigma <- values$sigma
```

We may then calculate the squared bias $\mathrm{E}(\tilde{f}(\mathbf{x}_0) - f(\mathbf{x}_0))^2$, using Equation (3). Plotting this for $\lambda \in [0, 2]$, we get the result as in Figure 1. We see that the squared bias increases with large $\lambda > 0.5$, and has a minimum when $\lambda = 0$ and $\lambda \approx 0.5$. This is expected for $\lambda = 0$, because we then get that $\tilde{\boldsymbol{\beta}}$ is equal to the OLS estimator. The bias measures how good the estimator is in estimating the real value, so this makes sense. From the figure it also appears that for $\lambda \approx 0.5$, the estimator $\tilde{\boldsymbol{\beta}}$ us estimating the real value $\boldsymbol{\beta}$ very good. For increasing λ after this, the estimator seems to do a worse job in estimating $\boldsymbol{\beta}$.

```
bias <- function(lambda, X, x0, beta) {
    p <- ncol(X)
    inv <- solve(t(X) %*% X + lambda * diag(p))
    value <- (t(x0) %*% inv %*% 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))
```

e)

We are now interested in $\mathrm{Var}(\tilde{f}(\mathbf{x}_0))$, which can be calculated using Equation (2). Using $\lambda \in [0, 2]$, the result is plotted in Figure 2. We notice that the variance is decreasing for increasing values of λ. That is, as long as

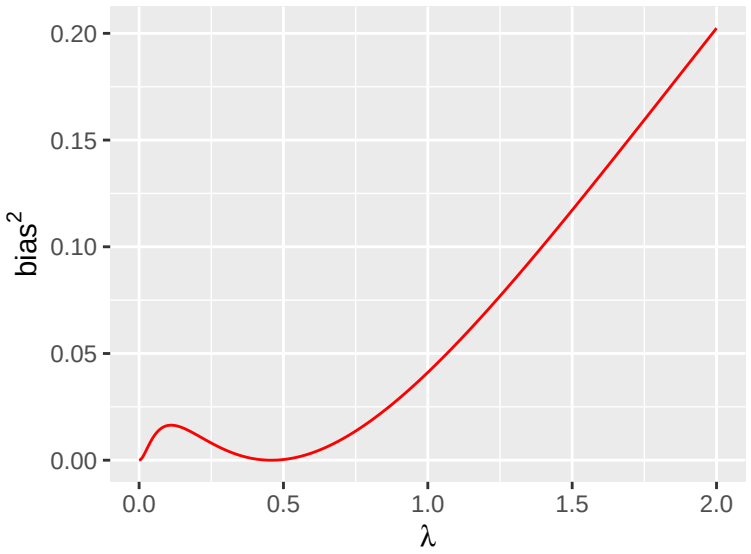


Figure 1: The squared bias $\mathrm{E}(\tilde{f}(\mathbf{x}_0) - f(\mathbf{x}_0))^2$ as a function of λ.

λ increases, the model is less prone to overfitting. Letting $\lambda \to \infty$, we also see that the variance tends to zero.

```
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") + theme(plot.title = element_text(hjust = 0.5))
```

f)

Lastly, we are interested in the expected MSE at $\mathbf{x}_0$, $\mathrm{E}((y_0 - \tilde{f}(\mathbf{x}_0))^2)$, which, when we know the squared bias and the variance from **e)**, is given as $(\text{bias})^2 + \mathrm{Var}(\tilde{f}(\mathbf{x}_0)) + \sigma^2$, because the irreducible error is σ^2. The plot of the expected MSE, the squared bias and the variance is shown in Figure 3, and by using `lambdas[which.min(exp_mse)]` we find that the minimal expected MSE is found when $\lambda \approx 0.993988$. That is, it is possbile to move to $\lambda > 0$, and taking a little bias, and reducing the variance, leading to a reduction in the expected MSE.

```
exp_mse <- BIAS + VAR + sigma^2

cols <- c(exp_mse = "blue", bias = "red", variance = "green4")
dfAll <- data.frame(lambda = lambdas, bias = BIAS, var = VAR, exp_mse = exp_mse)
```

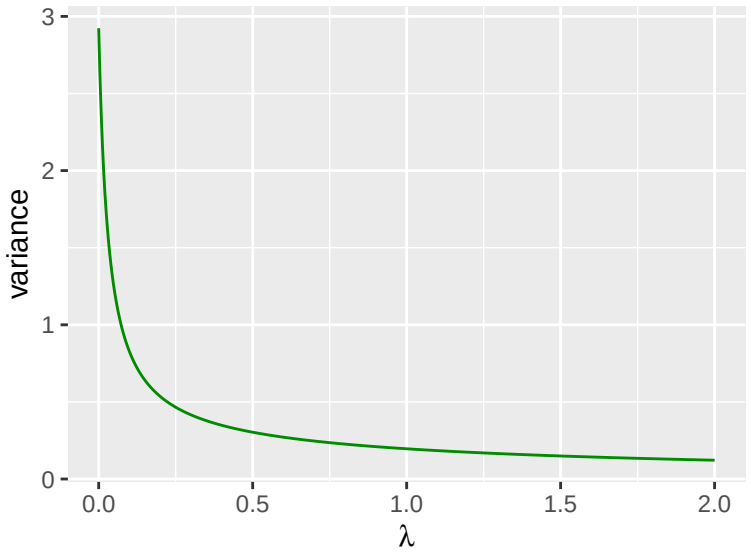


Figure 2: The variance $\mathrm{Var}(\tilde{f}(\mathbf{x}_0))$ as a function of λ.

```
ggplot(dfAll) + geom_line(aes(x = lambda, y = exp_mse, color = "exp_mse")) + geom_line(aes(x = lambda,
    y = bias, color = "bias")) + geom_line(aes(x = lambda, y = var, color = "variance")) +
    xlab(expression(lambda)) + ylab(expression(E(MSE))) + theme(legend.title = element_blank()) +
    scale_color_manual(values = cols)
```

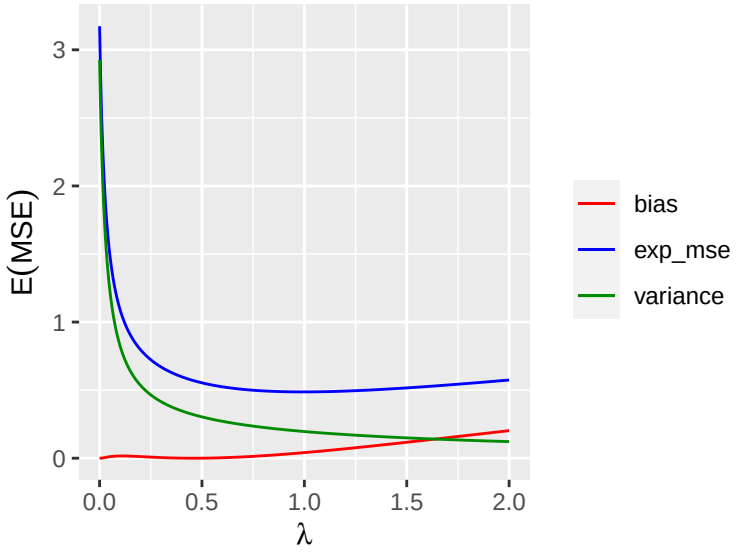


Figure 3: The expected MSE $\mathrm{E}((y_0 - \tilde{f}(\mathbf{x}_0))^2)$ as a function of λ, together with the squared bias and the variance.

Problem 2

Read the file:

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

a)

The number of deceased and non-deceased is calculated through a group by function on deceased followed by a count aggregate. We see that the number of deceased is 105 and the number of non-deceased is 1905.

```
deceased_count <- d.corona %>% group_by(deceased) %>% summarise(n()) %>% rename(c(count = "n()"))

print(deceased_count)
```

```
## # A tibble: 2 x 2
##   deceased count
##      <int> <int>
## 1        0  1905
## 2        1   105
```

For each country the number of males and females is calculated through a group by function on country and sex, followed by a count aggregate. We see that there is no major overrepresentation of either males or females in any country.

```
country_count_sex <- d.corona %>% group_by(country, sex) %>% summarise(n()) %>% pivot_wider(names_from =
    values_from = `n()`)

print(country_count_sex)
```

```
## # A tibble: 4 x 3
## # Groups:   country [4]
```

```
##   country   female  male
##   <fct>      <int> <int>
## 1 France        60    54
## 2 indonesia     30    39
## 3 japan        120   174
## 4 Korea        879   654
```

The number of deceased and non-deceased for each sex is calculated through a group by function on deceased and sex, followed by a count aggregate. We see that the fraction of deceased is higher amongst males than females in the sample.

```
sex_count_deceased <- d.corona %>% group_by(deceased, sex) %>% summarise(n()) %>%
    pivot_wider(names_from = deceased, values_from = `n()`) %>% rename(c(`non-deceased` = "0",
    deceased = "1"))

print(sex_count_deceased)
```

```
## # A tibble: 2 x 3
##   sex    `non-deceased` deceased
##   <fct>           <int>    <int>
## 1 female           1046       43
## 2 male              859       62
```

The number of deceased and non-deceased in France, separate for each sex, is calculated through first filtering out all rows which do not report on citizens of France, then grouping on deceased and sex followed by a count aggregate. We see again that the fraction of deceased is higher amongst males than females in the sample, also when confined to the samples of citizens of France.

```
sex_count_deceased_france <- d.corona %>% filter(country == "France") %>% group_by(deceased,
    sex) %>% summarise(n()) %>% pivot_wider(names_from = deceased, values_from = `n()`) %>%
    rename(c(`non-deceased` = "0", deceased = "1"))

print(sex_count_deceased_france)
```

```
## # A tibble: 2 x 3
##   sex    `non-deceased` deceased
##   <fct>           <int>    <int>
## 1 female             55        5
## 2 male               43       11
```

b)

We fit a logistic regression model to the data and print out the summary for reference when answering the questions.

```
deceased_logit = glm(deceased ~ sex + age + country, data = d.corona, family = binomial)
summary(deceased_logit)
```

```
##
## 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
```

(i)

We create a dataframe containing the values of the covariates, and use the predict function to get the probability:

```
korea_male_75 = data.frame(sex = "male", age = 75, country = "Korea")
p_korea_male_75 = predict(deceased_logit, newdata = korea_male_75, interval = "predict",
    type = "response")
```

The probability of a 75 year old Korean male to die from Covid is thus: 0.1084912

(ii)

We see that the model has a dummy variable sexmale with an estimated positive coefficient of 0.6260678. The p-value of this coefficient estimate is small. So there is evidence that males have higher probability of dying that females.

(iii)

We see that the model has dummy variables: countryindonesia, countryjapan and countryKorea, such that France serves as the base line for the model. The differences we observer are:

- Residents of Indonesia are found to have a smaller probability of dying from Covid than residents of France, but the finding is not significant (high p-value).
- Residents of Japan are found to have a smaller probability of dying from Covid than residents of France, and the finding is significant (very low p-value).
- Residents of Korea are found to have a smaller probability of dying from Covid than residents of France, and the finding is significant (low p-value).

In summary one must therefore conclude that there is evidence that country of residence influences the probability to decease.

(iv)

The estimated coefficient of age is 0.0271342 and it is found to be significant. We know that the log-odds is proportional to the linear expression of coefficients and covariates. Thus, increasing a persons age by 10 years would increase the log-odds by 0.2713421. The increase in odds would thus be by a factor: 1.3117238.

c)

(i)

To answer the question we fit a model using covariates: country, sex, age and the interaction sex * age.

```
deceased_sex_age_logit = glm(deceased ~ sex * age + country, data = d.corona, family = binomial)
summary(deceased_sex_age_logit)
```

```
##
## 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
```

We see that the p-value of the coefficient sexmale:age has a very high p-value. So there is no evidence for age being a greater risk factor for males compared to females.

(ii)

To answer the question we fit a model using covariates: sex, country, age and the interaction country * age. It should not matter that data for Japan and Korea are used in the fitting of the model, as they will not affect the coefficients we are interested in.

```
deceased_country_age_logit = glm(deceased ~ country * age + sex, data = d.corona,
    family = binomial)
summary(deceased_country_age_logit)
```

```
##
## Call:
## glm(formula = deceased ~ country * age + sex, 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 ***
## 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                   0.06693    0.02124   3.151  0.00163 **
## sexmale               0.62777    0.21056   2.981  0.00287 **
## countryindonesia:age -0.07189    0.03310  -2.172  0.02986 *
## countryjapan:age     -0.04630    0.02668  -1.736  0.08260 .
## countryKorea:age     -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
```

We find some evidence for age being a smaller risk factor for residents of Indonesia than for residents of France. The p-value is around 0.03. Further, we see that age is a smaller risk factor for residents of Japan and Korea compared to residents of France, although these results are not as significant. Since all three comparisons speak to residents of France having a higher risk factor (to varying strengths), and we have not made too many trials here, we conclude that there is some evidence for age being a smaller risk factor for residents of Indonesia than for residents of France.

d)

True, True, True, False.

Problem 3

```r
# read file
id <- "1i1cQPeoLLC_FyAHOnnqCnnrSBpn05_hO"  # google file ID
diab <- dget(sprintf("https://docs.google.com/uc?id=%s&export=download", id))
t = MASS::Pima.tr2
train = diab$ctrain
test = diab$ctest
```

a)

```r
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
```

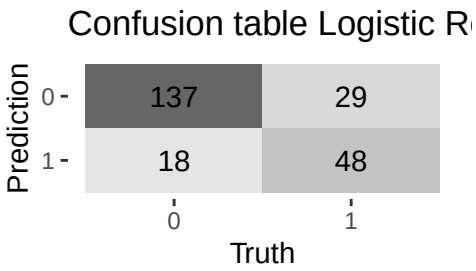
(i)

For simplicity we write $\beta_0 + \beta_1 x_{i1} + \cdots + \beta_7 x_{i7} = \mathbf{x}^\top \boldsymbol{\beta}$, where $\boldsymbol{\beta} = (\beta_0 \ \cdots \ \beta_7)^\top$ and $\mathbf{x} = (1 \ x_1 \ \cdots \ x_7)^\top$. Furthermore let $P(y_i = 1 | \mathbf{X} = \mathbf{x}_i) = p_i$. Starting from the expression for p_i we get

$$
\begin{aligned}
p_i &= \frac{e^{\mathbf{x}^\top \boldsymbol{\beta}}}{1 + e^{\mathbf{x}^\top \boldsymbol{\beta}}} \\
p_i + p_i e^{\mathbf{x}^\top \boldsymbol{\beta}} &= e^{\mathbf{x}^\top \boldsymbol{\beta}} \\
p_i &= e^{\mathbf{x}^\top \boldsymbol{\beta}} (1 - p_i) \\
\frac{p_i}{1 - p_i} &= e^{\mathbf{x}^\top \boldsymbol{\beta}} \\
\log\left(\frac{p_i}{1 - p_i}\right) &= \mathbf{x}^\top \boldsymbol{\beta} = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_7 x_{i7}. \quad \blacksquare
\end{aligned}
$$

(ii)

```r
logReg.prob = predict(logReg, newdata = test, type = "response")
logReg.pred = ifelse(logReg.prob > 0.5, "1", "0")
c = table(predicted = logReg.pred, true = test$diabetes)

cm = conf_mat(c)

autoplot(cm, type = "heatmap") + ggtitle("Confusion table Logistic Regression")
```



From the table above we find sensitivity $= \frac{48}{29+48} = \frac{48}{77} = 0.623$, and specificity $= \frac{137}{137+18} = \frac{137}{155} = 0.884$.

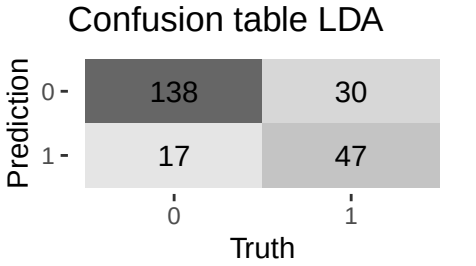
b)

(i)

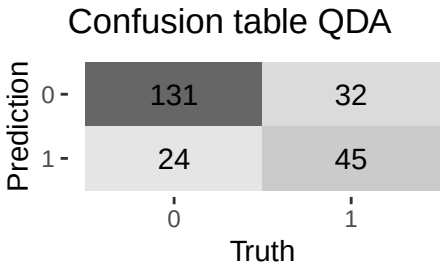
π_k is the prior probability for class k, i.e $P(Y = k)$, so for the diabetes problem, π_1 is the probability for a random woman of having diabetes, while π_0 is the probability of not having diabetes. $\boldsymbol{\mu}_k$ is the mean vector for each class, so in this case μ_1 and μ_0 will be the mean values of each predictors for a correspondingly diabetic or non-diabetic person. $\boldsymbol{\Sigma}$ is the covariance matrix between the predictors and is in LDA assumed to be the same for all classes. $f_k(x)$ is the conditional distribution of the predictors of each class, and is assumed to be normal distribution, with mean $\boldsymbol{\mu_k}$ and covariance matrix $\boldsymbol{\Sigma}$.

(ii)

```
lda.model = lda(diabetes ~ ., data = train)
qda.model = qda(diabetes ~ ., data = train)

lda.modelProb = predict(lda.model, newdata = test)$posterior
lda.modelPred = predict(lda.model, newdata = test)$class
lc = table(predicted = lda.modelPred, true = test$diabetes)

qda.modelProb = predict(qda.model, newdata = test)$posterior
qda.modelPred = predict(qda.model, newdata = test)$class
qc = table(predicted = qda.modelPred, true = test$diabetes)

cml = conf_mat(lc)
cmq = conf_mat(qc)

autoplot(cml, type = "heatmap") + ggtitle("Confusion table LDA")
```




```
autoplot(cmq, type = "heatmap") + ggtitle("Confusion table QDA")
```



The difference between LDA and QDA models, is that in LDA, one assumes that the covariance matrices is the same for all classes ($\Sigma_k = \Sigma$). By cancellations this leads to a linear decision boundary (discriminant function):

$$\delta_k(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \log \pi_k.$$

In QDA one does not assume that the covariance matrices is the same for all classes, and thus we do not obtain the same cancellation. This gives us a quadratic decision boundary

$$\delta_k(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^\top \boldsymbol{\Sigma}_k^{-1}\mathbf{x} + \mathbf{x}^\top \boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| + \log \pi_k.$$

c)

(i)

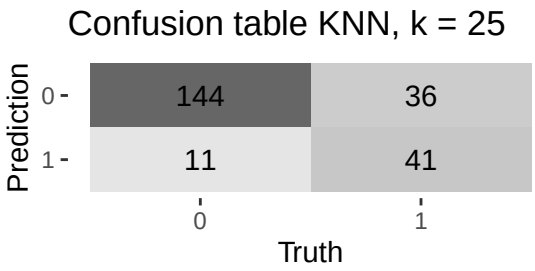
When classifying a new observation in the KNN approach, one first identifies the k nearest points in the training data, measured in the euclidean metric. Then the observation is classified to the majority class amongst those k nearest points, i.e. classified to the class with the highest proportion of members among the k points in question.

(ii)

The tuning parameter k can be chosen by looking at the test error (CV-error, if using cross-validation) for a range of different values of k. Then choose the value that gives the lowest error.

(iii)

```
knnMod = knn(train = train, test = test, cl = train$diabetes, k = 25, prob = T)
kc = table(knnMod, test$diabetes)

cmk = conf_mat(kc)
autoplot(cmk, type = "heatmap") + ggtitle("Confusion table KNN, k = 25")
```



We then get: Sensitivity $= \frac{41}{36+41} = \frac{41}{77} = 0.532$ and specificity $= \frac{144}{144+11} = \frac{144}{155} = 0.929$.

d)

```
knnModProb = attributes(knnMod)$prob
knnModProb = ifelse(knnMod == 0, 1 - knnModProb, knnModProb)

glmroc = roc(response = test$diabetes, predictor = logReg.prob)
ldaroc = roc(response = test$diabetes, predictor = lda.modelProb[, 2])
qdaroc = roc(response = test$diabetes, predictor = qda.modelProb[, 2])
knnroc = roc(response = test$diabetes, predictor = knnModProb)

dat = data.frame(diabetes = test$diabetes, glm = logReg.prob, lda = lda.modelProb[,
    2], qda = qda.modelProb[, 2], knn = knnModProb)
dat_long = melt_roc(dat, "diabetes", c("glm", "lda", "qda", "knn"))
ggplot(dat_long, aes(d = D, m = M, color = name)) + geom_roc(n.cuts = F, size = 0.5) +
    xlab("1-Specificity") + ylab("Sensitivity")
```

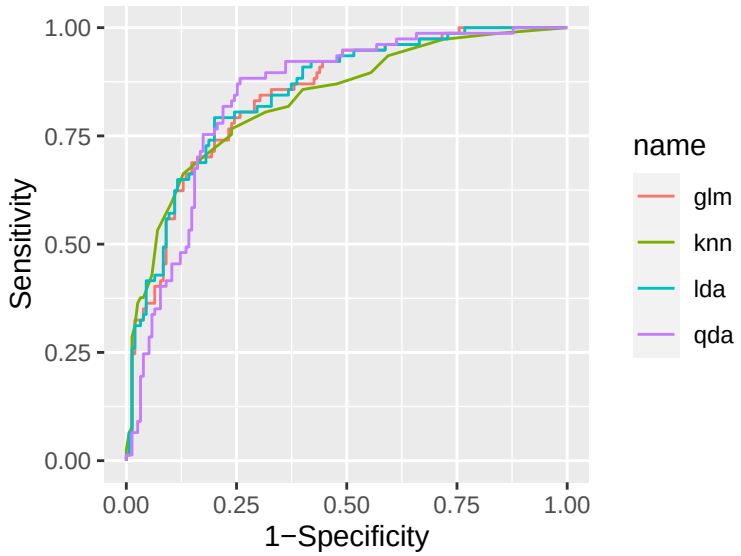


Figure 4: ROC curve for Logistic Regression, LDA, QDA and KNN, measured on the diabetes data set.

```
auc(glmroc)
```

```
## Area under the curve: 0.8451
```

```
auc(ldaroc)
```

```
## Area under the curve: 0.849
```

```
auc(qdaroc)
```

```
## Area under the curve: 0.8415
```

```
auc(knnroc)
```

```
## Area under the curve: 0.8334
```

We observe that the LDA model has the highest AUC value, 0.849, followed by the Logistic Regression model, AUC = 0.8451. Thus measured in the AUC metric, we would say that LDA performs the best. Since the difference in AUC value between Logistic Regression and LDA is relatively small, we would prefer Logistic Regression for interpretability.

Problem 4

a)

We wish to show that for the linear regression model, the LOOCV statistic is given by

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

where $h_i = \mathbf{x}_i^\top (X^\top X)^{-1}\mathbf{x}_i$, and $\mathbf{x}_i^\top$ is the i-th row of X.

Generally we can write $\text{CV} = \sum_{i=1}^{N} e_{(-i)}^2/N$, where $e_{(-i)} = y_i - \hat{y}_{(-i)}$. We use the notation $A_{(-i)}$ to symbolize that element i is removed from A if it is a vector, and row i is removed from A if it is a matrix. For a linear regression model $\mathbf{Y} = X\boldsymbol{\beta} + \varepsilon$, the estimate of $\boldsymbol{\beta}$ without the i-th case is

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

From what we then know, we may write $e_{(-i)} = y_i - \mathbf{x}_i^\top \hat{\boldsymbol{\beta}}_{(-i)}$. We want another expression for $\hat{\boldsymbol{\beta}}_{(-i)}$, and using the Sherman-Morrison formula,

$$(X_{(-i)}^\top X_{(-i)})^{-1} = (X^\top X - \mathbf{x}_i\mathbf{x}_i^\top)^{-1} = (X^\top X)^{-1} + \frac{(X^\top X)^{-1}\mathbf{x}_i\mathbf{x}_i^\top (X^\top X)^{-1}}{1 - \mathbf{x}_i (X^\top X)^{-1}\mathbf{x}_i^\top}.$$

By the definition of h_i, we then get that

$$(X_{(-i)}^\top X_{(-i)})^{-1} = (X^\top X)^{-1} + \frac{(X^\top X)^{-1}\mathbf{x}_i\mathbf{x}_i^\top (X^\top X)^{-1}}{1 - h_i}.$$

It is also clear that $X_{(-i)}^\top \mathbf{Y}_{(-i)} = X^\top \mathbf{Y} - \mathbf{x}_i y_i$, and thus

$$\hat{\boldsymbol{\beta}}_{(-i)} = (X_{(-i)}^\top X_{(-i)})^{-1} X_{(-i)}^\top \mathbf{Y}_{(-i)} = \left[(X^\top X)^{-1} + \frac{(X^\top X)^{-1}\mathbf{x}_i\mathbf{x}_i^\top (X^\top X)^{-1}}{1-h_i}\right](X^\top \mathbf{Y} - \mathbf{x}_i y_i).$$

Multiplying out this expression we then get

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

where we let $e_i = y_i - \hat{y}_i$. This allows us to find

$$e_{(-i)} = y_i - \mathbf{x}_i^\top \hat{\boldsymbol{\beta}}_{(-i)} = y_i - \mathbf{x}_i^\top \hat{\boldsymbol{\beta}} + \frac{\mathbf{x}_i^\top (X^\top X)^{-1}\mathbf{x}_i}{1-h_i} e_i = e_i + \frac{h_i}{1-h_i} e_i = \frac{e_i}{1-h_i}.$$

It then follows that

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

which was what to be shown. ***Q.E.D.***

b)

False, True, True, False.

Problem 5

Read the file:

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

a)

```
bodyfat_lm = lm(d.bodyfat$bodyfat ~ d.bodyfat$age + d.bodyfat$weight + d.bodyfat$bmi)
```

The R^2-value is: 0.5803041

b)

(i)

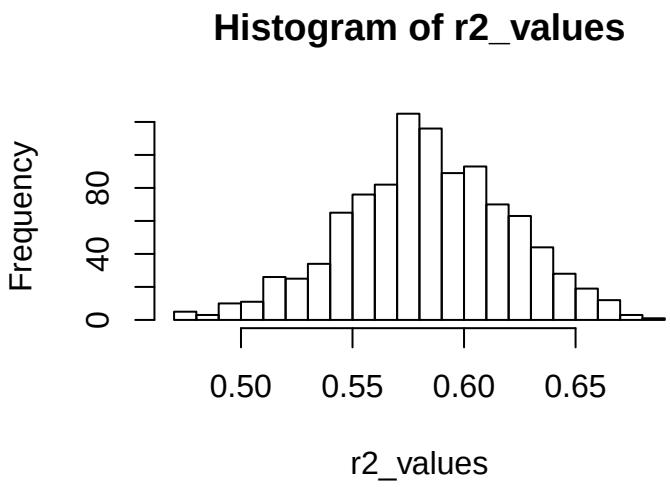
An implementation of bootstrapping 1000 R^2-values by using samples from the bodyfat dataset:

```
bootstrap_R2 <- function(data, n_samples) {

    indecies <- c(1:n_samples)
    r2_values <- c(1:n_samples)

    for (i in indecies) {
        samples <- data[sample(nrow(data), size = nrow(data), replace = TRUE), ]
        samples_lm <- lm(samples$bodyfat ~ samples$age + samples$weight + samples$bmi)
        r2_values[i] <- summary(samples_lm)$r.squared
    }
    return(r2_values)
}

set.seed(4268)
n_bootstrap_samples <- 1000
r2_values <- bootstrap_R2(d.bodyfat, 1000)
```

(ii)

A histogram of the R^2 values sampled:

```
hist(r2_values, breaks = 20)
```



(iii)

```
est_mean <- mean(r2_values)
rse <- (sum((est_mean - r2_values)^2)/(n_bootstrap_samples - 1))^0.5
```

The estimated mean is: 0.5833966 and residual standard error is: 0.03705. A 95% confidence interval is therefore: [0.5107785, 0.6560146].

(iv)

We see that the fraction of variance explained by the model is estimated to be: 0.5803041. We have used bootstrapping to find the standard error to be 0.03705. The point estimate of R^2 is therefore uncertain. With 95% probability we have estimated that the fraction of variance explained by the model will lie in the interval [0.5107785, 0.6560146].