Compulsory exercise 1: Group 39

TMA4268 Statistical Learning V2021

Alexander J Ohrt, Jim Totland

22 February, 2021

Problem 1

a)

By extending the given univariate regression problem to a multivariate regression problem that allows for several observations, we have that $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$. Hence, we have $\mathsf{E}(\mathbf{Y}) = \mathsf{E}(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}) = \mathbf{X}\boldsymbol{\beta}$ and $\mathsf{Cov}(\mathbf{Y}) = \sigma^2 I$, assuming that $\boldsymbol{\varepsilon} \sim N(0, \sigma^2 I)$. Then

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

and

$$\begin{aligned}\mathsf{Cov}(\widetilde{\boldsymbol{\beta}}) &= \mathsf{Cov}((\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}) = ((\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T)\mathsf{Cov}(\mathbf{Y})((\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T)^T \\ &= ((\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T)\sigma^2 I\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-T} = \sigma^2((\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-T}) \\ &= \sigma^2((\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}),\end{aligned}$$

where we have used that $(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-T}$ in the last equality, because $\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}$ is symmetric.

In both these equations it is apparent that the moments are equal to those of the OLS estimator when $\lambda = 0$.

b)

The requested moments of $\widetilde{f}(\mathbf{x}_0) = \mathbf{x}_0^T\widetilde{\boldsymbol{\beta}}$ are

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

and

$$\begin{aligned}\mathsf{Cov}(\widetilde{f}(\mathbf{x}_0)) &= \mathsf{Cov}(\mathbf{x}_0^T\widetilde{\boldsymbol{\beta}}) = \mathbf{x}_0^T\mathsf{Cov}(\widetilde{\boldsymbol{\beta}})\mathbf{x}_0 \\ &= \sigma^2\mathbf{x}_0^T((\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1})\mathbf{x}_0.\end{aligned}$$

c)

The expected MSE at $\mathbf{x}_0$ is

$$\begin{aligned}
\mathsf{E}[(y_0 - \widetilde{f}(\mathbf{x}_0))^2] &= [\mathsf{E}(\widetilde{f}(\mathbf{x}_0) - f(\mathbf{x}_0))]^2 + \mathsf{Var}(\widetilde{f}(\mathbf{x}_0)) + \mathsf{Var}(\varepsilon) \\
&= [\mathsf{E}(\widetilde{f}(\mathbf{x}_0)) - \mathsf{E}(f(\mathbf{x}_0))]^2 + \mathsf{Cov}(\widetilde{f}(\mathbf{x}_0)) + \sigma^2 \\
&= [\mathbf{x}_0^T(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}\beta - \mathbf{x}_0^T\beta]^2 + \sigma^2\mathbf{x}_0^T((\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1})\mathbf{x}_0 + \sigma^2.
\end{aligned}$$

Since there is no obvious way to simplify this, it will be left like this. This is also practical since it is easy to distinguish between the irreducible error, the variance of prediction and the squared bias.

d)

```r
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
```

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

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

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

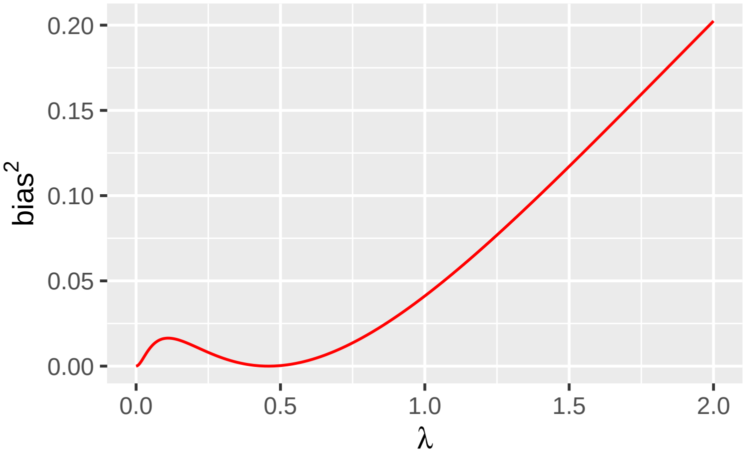
```
#> [1] 81  1
```

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

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

```r
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))
```

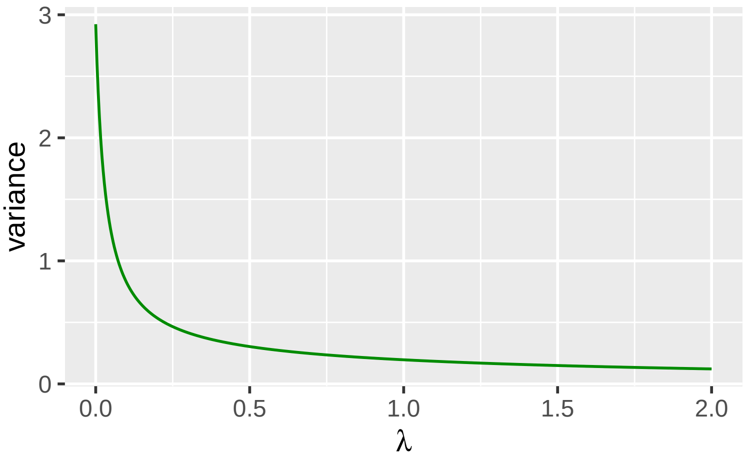


The graph shows that the bias of the ridge regression estimator increases as λ grows. OLS is unbiased, so, as expected, the bias is zero when $\lambda = 0$. Note that $\lambda \approx 0.5$ appears to be a sweet spot for this estimator when comparing all $\lambda > 0$, since the bias is low there. Perhaps this can be useful later.

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")
```



The variance begins at ≈ 2.923 and decreases with λ. Hence, it is apparent that the ridge regression estimator is advantageous, when looking at solely variance, compared to the OLS estimator, since the variance is decreasing with λ. Despite this, when adding the bias and the variance, the OLS estimator may still have a lower expected MSE than the ridge regression estimator. Finally, note that the changes in the variance are larger for $\lambda \in [0, 2]$ than the changes in the bias (compared to the plot in task d)), which indicates that the variance dominates the change in expected MSE for the ridge regression estimation.

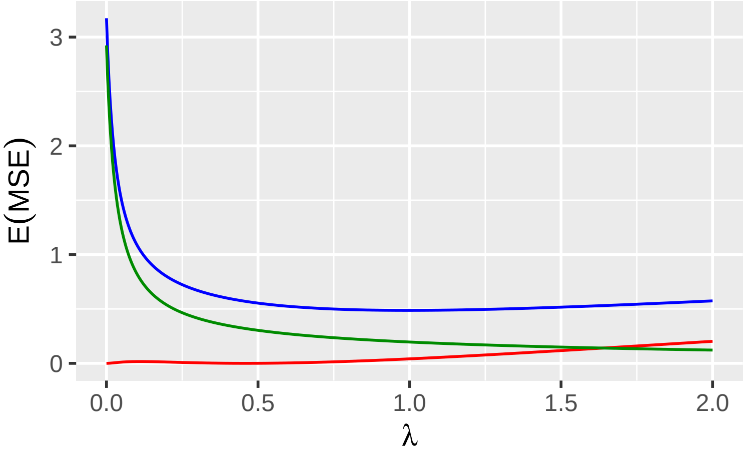
f)

```
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)))
```



Now we are able to conclude that the ridge regression estimator has a lower expected MSE compared to the OLS estimator, as gathered from the blue line in the plot. The optimal value of λ, which minimizes MSE, is ≈ 0.994. Hence, despite the fact that the bias is higher for the ridge regression estimator, the total expected MSE is lower, since the decrease in variance counters the increase in bias.

Problem 2

```
# Read 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 tables are reported below.

```
table1 <- summarise(d.corona, deceased = sum(d.corona$deceased),
                   non_deceased = nrow(d.corona) - deceased)
knitr::kable(table1)
```

deceased	non_deceased
105	1905

```r
table2 <- table(d.corona$country, d.corona$sex)
knitr::kable(table2)
```

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

```r
table3 <- table(d.corona$deceased, d.corona$sex)
rownames(table3) = c("non-deceased", "deceased")
knitr::kable(table3)
```

	female	male
non-deceased	1046	859
deceased	43	62

```r
d.france <- filter(d.corona, country == "France")
table4 <- table(d.france$deceased, d.france$sex)
rownames(table4) = c("non-deceased", "deceased")
knitr::kable(table4)
```

	female	male
non-deceased	55	43
deceased	5	11

b)

Since we want to understand the probability of decease of covid-19, we fit a logistic regression model as opposed a linear regression model.

```r
glm.fit <- glm(deceased ~ ., family = "binomial", data = d.corona)
summary(glm.fit)
```

```
#> 
#> Call:
#> glm(formula = deceased ~ ., 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) The probability to die of covid for a male age 75 living in Korea can be predicted from the model. The prediction is found by

```
x0 <-  data.frame(sex = "male", age = 75, country = "Korea")
pred <- predict(glm.fit, newdata = x0, type = "response")
pred
```

```
#>         1 
#> 0.1084912
```

Hence, the probability to die of covid for a male age 75 living in Korea is ≈ 0.108.

(ii) The p-value associated with `sexmale` is relatively small, and since the coefficient is positive, this constitutes some evidence that males have a higher probability of dying.

(iii) Yes. Both the `countryjapan` and `countryKorea` coefficients have relatively low p-values and are negative, which could be used as evidence that the probability of decease is lower in Japan and Korea compared to the reference category, France. `countryIndonesia` is not significant (large p-value), so there is no evidence that the probability of decease is any higher in Indonesia than in France.

(iv) Since we have used logistic regression, the predicted odds of decease, given an observation $\boldsymbol{x}$, is

$$\frac{p(\boldsymbol{x})}{1-p(\boldsymbol{x})} = e^{\boldsymbol{x}^T\hat{\boldsymbol{\beta}}}.$$

Thus, the odds of decease increases by a factor of $e^{10\hat{\beta_{\text{age}}}} \approx 1.312$ when `age` increases by 10 and all other covariates are held constant.

c)

```
log.fit1 <- glm(deceased ~. + sex:age, data = d.corona, family="binomial")
summary(log.fit1)
```

```
#> 
#> Call:
#> glm(formula = deceased ~ . + sex:age, 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
```

(i) As seen above, we have fitted the full logistic regression with an interaction term between `age` and `sex`. As the summary shows, the interaction effect between `sexmale` and `age` is not significant, since the p-value is large. Hence, although the coefficient is slightly positive, we cannot conclude that age is a greater risk factor for men.

```
log.fit2 <- glm(deceased ~ . + age:country, data = d.corona, family="binomial")
summary(log.fit2)
```

```
#>
#> Call:
#> glm(formula = deceased ~ . + 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
```

(ii) As seen above, we have fitted the full logistic regression with an interaction term between `age` and `country`. Since `France` is the reference level, and the interaction-term coefficient `age:countryindonesia` has the value ≈ -0.072 with a somewhat low p-value, we can infer that age is a greater risk factor for the French population than for the Indonesian population.

d)

TRUE, TRUE, TRUE, FALSE.

Problem 3

```r
# Read file.
id <- "1i1cQPeoLLC_FyAH0nnqCnnrSBpn05_h0" # 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) We have that

$$p_i = \frac{e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_7 x_{i7}}}{1+e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_7 x_{i7}}} := \frac{e^{\eta_i(x)}}{1+e^{\eta_i(x)}} \implies \frac{p_i}{1-p_i} = \frac{\frac{e^{\eta_i(x)}}{1+e^{\eta_i(x)}}}{1-\frac{e^{\eta_i(x)}}{1+e^{\eta_i(x)}}} = e^{\eta_i(x)}$$

$$\implies \mathsf{logit}(p_i) = \mathsf{log}\left(\frac{p_i}{1-p_i}\right) = \eta_i(x) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_7 x_{i7}$$

(ii) The classification is done below.

```r
glm.probs <- predict(logReg, newdata = test, type = "response")
glm.preds <- ifelse(glm.probs > 0.5, 1, 0)
conf.table.glm <- table(predicted = glm.preds, true = test$diabetes)
conf.table.glm
```

```
#>          true
#> predicted   0   1
#>         0 137  29
#>         1  18  48
```

The sensitivity is $48/(48+29) \approx 0.623$ and the specificity is $137/(137+18) \approx 0.884$.

b)

(i) π_k is the prior probability for an observation, given by $\pi_k = P(y=k)$. $f_k(\mathbf{x})$ is the gaussian pdf for class k, with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$. In LDA $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma} \quad \forall k$, whereas in QDA each $\boldsymbol{\Sigma}_k$ are assumed to be class-specific. In this case, the class 1, i.e. presence of diabetes, has the mean vector $\boldsymbol{\mu}_1$ and the class 0, i.e. non-presence of diabetes, has the mean vector $\boldsymbol{\mu}_0$. $\boldsymbol{\Sigma}$ is the covariance matrix of both classes, since we are performing LDA. Hence, $f_1(\mathbf{x})$ is the gaussian pdf for class 1, with mean $\boldsymbol{\mu}_1$ and covariance matrix $\boldsymbol{\Sigma}$, while $f_0(\mathbf{x})$ is the gaussian pdf for class 0, with mean $\boldsymbol{\mu}_0$ and covariance matrix $\boldsymbol{\Sigma}$.

(ii) The fits are seen below

```r
lda.diabetes <- lda(diabetes~., data = train)
qda.diabetes <- qda(diabetes~., data = train)

# Only need the prob if diabetes is present.
lda.diabetes.probs <- predict(lda.diabetes, newdata = test)$posterior[, 2]
lda.preds <- ifelse(lda.diabetes.probs > 0.5, 1, 0)
conf.table.lda.diabetes <- table(predicted = lda.preds, true = test$diabetes)
conf.table.lda.diabetes
```

```
#>          true
#> predicted   0   1
#>         0 138  30
#>         1  17  47
```

```r
# Only need the prob if diabetes is present.
qda.diabetes.probs <- predict(qda.diabetes, newdata = test)$posterior[, 2]
qda.preds <- ifelse(qda.diabetes.probs > 0.5, 1, 0)
conf.table.qda.diabetes <- table(predicted = qda.preds, true = test$diabetes)
conf.table.qda.diabetes
```

```
#>          true
#> predicted   0   1
#>         0 131  32
#>         1  24  45
```

The sensitivity and specificity for LDA are thus $47/(47+30) \approx 0.61$ and $138/(138+17) \approx 0.89$, respectively. The sensitivity and specificity of QDA are $45/(45+32) \approx 0.584$ and $131/(131+24) \approx 0.845$, respectively.

The difference between the methods is that the covariance matrix is assumed to be equal across all classes (in this case: both classes) in LDA, which gives linear discriminant functions and a linear decision boundary, while in QDA the covaraince matrix is assumed to be class-specific, which gives a quadratic discriminant function for each class and a quadratic decision boundary.

c)

(i) In the KNN approach, a new observation is classified by using the k nearest points (in Euclidean distance) to the observation in question. The new observation is classified to the most occuring class among the k nearest points.
(ii) We would choose the tuning parameter k based on a κ-fold cross validation, with $\kappa = 5$ or $\kappa = 10$.
(iii) The KNN classification fit can be seen below

```
set.seed(123) # For reproducibility, e.g. in case of ties.
knn.diabetes <- knn(train = train, test = test, cl = train$diabetes, k=25, prob=T)
conf.table.knn <- table(predicted = knn.diabetes, true = test$diabetes)
conf.table.knn
```

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

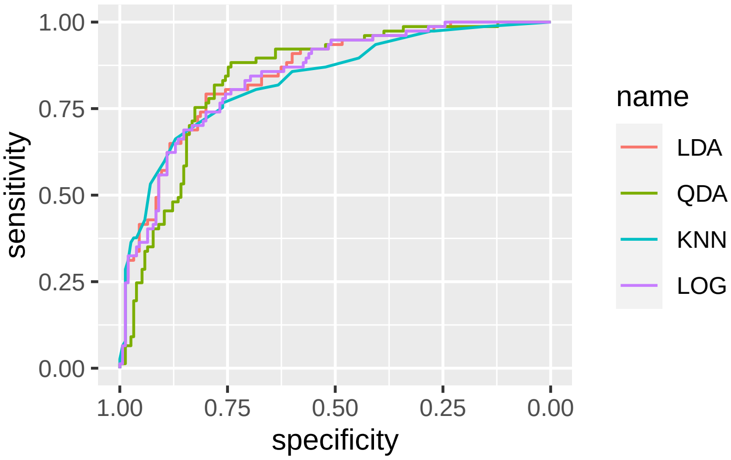
The sensitivity is $41/(41+36) \approx 0.532$ and the specificity is $144/(144+11) \approx 0.929$.

d) ROC curves

```
knn.probs <- ifelse(knn.diabetes == 0, 1 - attributes(knn.diabetes)$prob, attributes(knn.diabetes)$prob
glm.log.probs <- predict(logReg, newdata = test, type="response")

ldaroc = roc(response = test$diabetes, predictor = lda.diabetes.probs, direction = "<")
qdaroc = roc(response = test$diabetes, predictor = qda.diabetes.probs, direction = "<")
knnroc = roc(response = test$diabetes, predictor = knn.probs, direction = "<")
glm.logroc = roc(response = test$diabetes, predictor = glm.log.probs, direction = "<")


ggroc(list(LDA  = ldaroc, QDA = qdaroc, KNN = knnroc, LOG = glm.logroc))
```



The area under the ROC curve is 0.849 for LDA, 0.841 for QDA, 0.833 for KNN and 0.845 for logistic regression (LOG). We see that LDA preforms the best with respect to the AUC, which indicates that a linear decision boundary is appropriate. Thus, if the task is to create an interpretable model, we would choose the logistic regression, since it is more interpretable than LDA with its readily available coefficients and associated formulas for probability and odds.

Problem 4

a)

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

This gives

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

which gives

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

which completes the proof.

b)

FALSE, TRUE, FALSE, FALSE.

Problem 5

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

a)

```r
lm.fit5 <- lm(bodyfat ~ age + weight + bmi, data = d.bodyfat)
```

The R^2 is ≈ 0.5803.

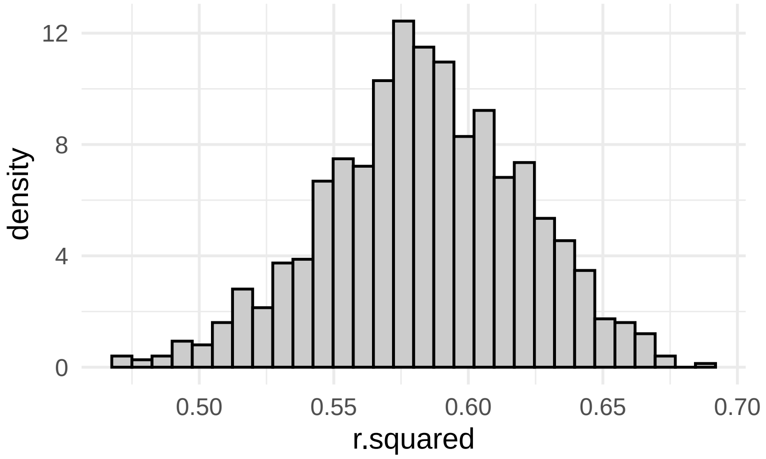
b)

```r
set.seed(4268)

boot.fn <-  function(data, index) {
  return(summary(lm(bodyfat ~ age + weight + bmi, data = data, subset = index))$r.squared)
}

B <- 1000
r.squared <- rep(NA, B)
for (b in 1:B){
  r.squared[b] <- boot.fn(d.bodyfat, sample(nrow(d.bodyfat), nrow(d.bodyfat), replace = T))
}

df = data.frame(r.squared = r.squared, norm_den = dnorm(r.squared, mean(r.squared),
                                                         sd(r.squared)))
ggplot(df) + geom_histogram(aes(x = r.squared, y = ..density..), fill = "grey80",color = "black") +
   theme_minimal()
```



```r
sd(r.squared)
```

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

```r
quantile(r.squared, c(0.025, 0.975))
```

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

The standard error is 6.385 % of R^2. Also, R^2 is contained in the confidence interval. This suggests that the bootstrap can be used to estimate the uncertainty of R^2. Hva mer skal vi skrive i tolkningen? Også: Ikke 12 sider!