# Compulsory exercise 1: Group 13

TMA4268 Statistical Learning V2019

Erik Andrè Klepp Vik and Vemund Tjessem

07 februar, 2020

## Problem 1

For this problem you will need to include some LaTex code. Please install latex on your computer and then consult Compulsor1.Rmd for hints how to write formulas in LaTex.

### a)

The expression for the test mean squared error is

$$\text{MSE}_{\text{test}} = \frac{1}{n_0}\sum_{j=1}^{n_0}(y_{0j} - \hat{f}(x_{0j}))^2 \tag{1}$$

The expected test mean squared error (MSE) at $x_0$ is defined as:

$$\text{E}[y_0 - \hat{f}(x_0)]^2 \tag{2}$$

### b)

Using the fact that $y_0 = f(x_0) + \varepsilon$ gives

$$\text{E}[y_0 - \hat{f}(x_0)]^2 = \underbrace{\text{Var}(\varepsilon)}_{\text{Irreducible error}} + \underbrace{\text{Var}(\hat{f}(x_0))}_{\text{Variance of prediction}} + \underbrace{\Big(f(x_0) - \text{E}[\hat{f}(x_0)]\Big)^2}_{\text{Squared bias}} \tag{3}$$

### c)

### d)

### e)

### f)

### g)

## Problem 2

Here is a code chunk:

```
id <- "1nLen1ckdnX4P9n8ShZeU7zbXpLc7qiwt"  # google file ID
d.worm <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download", id))
head(d.worm)
```

```
##   Gattung Nummer GEWICHT FANGDATUM MAGENUMF
## 1      Oc     32    0.19  23.09.97     1.56
## 2      Oc     34    0.59  23.09.97     1.63
## 3      Oc     48    0.09  23.09.97     1.69
## 4      Oc     55    0.23  23.09.97     1.69
## 5      Oc     41    0.24  23.09.97     1.75
## 6      Oc     24    0.19  23.09.97     1.81
```

### a)

### b)

Below you have to complete the code and then replace `eval=FALSE` by `eval=TRUE` in the chunk options:

```
ggplot(d.worm, aes(x = ..., y = ..., colour = ...)) + geom_point() + theme_bw()
```

Note that the default figure width and height have been set globally as `fig.width=4, fig.height=3`, but if you would like to change that (e.g., due to space constraints), you can include a different width and height directly into the chunk options, again using `fig.width=..., fig.height=...`.

### c)

### d)

### e)

### f)

### g)

## Problem 3

Loading the files:

```
id <- "1GNbIhjdhuwPOBr0Qz82JMkdjUVBuSoZd"
tennis <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download", id),
    header = T)
head(tennis)
```

```
##         Player1      Player2 Result ACE.1 UFE.1 ACE.2 UFE.2
## 1     M.Koehler   V.Azarenka      0     2    18     3    14
## 2    E.Baltacha   F.Pennetta      0     0    10     4    14
## 3     S-W.Hsieh      T.Maria      1     1    13     2    29
## 4      A.Cornet       V.King      1     4    30     0    45
## 5  Y.Putintseva   K.Flipkens      0     2    28     6    19
## 6 A.Tomljanovic B.Jovanovski      0     6    42    11    40
```

### a)

We have

$$p_i = \frac{e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\beta_3 x_{i3}+\beta_4 x_{i4}}}{1+e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\beta_3 x_{i3}+\beta_4 x_{i4}}} \tag{4}$$

which gives

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \log\left(\frac{\frac{e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\beta_3 x_{i3}+\beta_4 x_{i4}}}{1+e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\beta_3 x_{i3}+\beta_4 x_{i4}}}}{1-\frac{e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\beta_3 x_{i3}+\beta_4 x_{i4}}}{1+e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\beta_3 x_{i3}+\beta_4 x_{i4}}}}\right) \tag{5}$$

Multiplying both the numerator and denominator in Equation 5 by $1+e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\beta_3 x_{i3}+\beta_4 x_{i4}}$ gives

$$\text{logit}(p_i) = \log\left(\frac{e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\beta_3 x_{i3}+\beta_4 x_{i4}}}{1+e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\beta_3 x_{i3}+\beta_4 x_{i4}}-\left(e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\beta_3 x_{i3}+\beta_4 x_{i4}}\right)}\right) \tag{6}$$

which further results in

$$\text{logit}(p_i) = \log(e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\beta_3 x_{i3}+\beta_4 x_{i4}}) = \beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\beta_3 x_{i3}+\beta_4 x_{i4} \tag{7}$$

It can be seen from Equation 7 that $\text{logit}(p_i)$ is a linear function of the covariates $x_{i1}$, $x_{i2}$, $x_{i3}$ and $x_{i4}$.

## b)

```
r.tennis = glm(Result ~ ACE.1 + ACE.2 + UFE.1 + UFE.2, data = tennis, family = "binomial")
summary(r.tennis)
```

```
##
## Call:
## glm(formula = Result ~ ACE.1 + ACE.2 + UFE.1 + UFE.2, family = "binomial",
##     data = tennis)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0517  -0.8454   0.3725   0.8773   2.0959
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.02438    0.59302  -0.041 0.967211
## ACE.1        0.36338    0.10136   3.585 0.000337 ***
## ACE.2       -0.22388    0.07369  -3.038 0.002381 **
## UFE.1       -0.09847    0.02840  -3.467 0.000527 ***
## UFE.2        0.09010    0.02479   3.635 0.000278 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 163.04  on 117  degrees of freedom
## Residual deviance: 124.96  on 113  degrees of freedom
## AIC: 134.96
##
## Number of Fisher Scoring iterations: 4
```

```
coeffs = coef(r.tennis)
eterm1 = exp(coeffs[1] + coeffs[2] + coeffs[3] + coeffs[4] + coeffs[5])
p1 = eterm1/(1 + eterm1)
eterm2 = exp(coeffs[1] + 2 * coeffs[2] + coeffs[3] + coeffs[4] + coeffs[5])
p2 = eterm2/(1 + eterm2)
rat = p2/p1
```

Since the $\beta_1$ is positive one more ace for player one would increase the probability of player 1 winning. It also makes sense that an ace for player 1 would increase the probability of player 1 winning. The propability increase is dependent on the other parameters as well, but assuming 1 ace and 1 unforced error for each player, one more ace for player 1 would increase the probability of player 1 winning by a ratio of 1.1685197.

## c)

```
# make variables for difference
tennis$ACEdiff = tennis$ACE.1 - tennis$ACE.2
tennis$UFEdiff = tennis$UFE.1 - tennis$UFE.2
```

```r
# divide into test and train set
n = dim(tennis)[1]
n2 = n/2
set.seed(1234)  # to reproduce the same test and train sets each time you run the code
train = sample(c(1:n), replace = F)[1:n2]
tennisTest = tennis[-train, ]
tennisTrain = tennis[train, ]

# Fitting a logistic regression model on the form Result ~ ACEdiff + UFEdiff on
# the traininng set
fit.3c = glm(Result ~ ACEdiff + UFEdiff, data = tennis, family = "binomial")
summary(fit.3c)
```

```
##
## Call:
## glm(formula = Result ~ ACEdiff + UFEdiff, family = "binomial",
##     data = tennis)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1923  -0.8922   0.3821   0.9089   2.1254
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.21576    0.21833   0.988    0.323
## ACEdiff      0.27273    0.06251   4.363 1.28e-05 ***
## UFEdiff     -0.09101    0.02285  -3.983 6.80e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 163.04  on 117  degrees of freedom
## Residual deviance: 126.48  on 115  degrees of freedom
## AIC: 132.48
##
## Number of Fisher Scoring iterations: 4
```

## d)

```r
x = 1:10
disp(x)
```

**e)**

**f)**

**g)**

**h)**

## Problem 4

**a)**

**b)**

**c)**

**d)**