Compulsory exercise 1: Group 6

TMA4268 Statistical Learning V2022

August Arnstad, Markus Stokkenes and Ulrik Unneberg

21 februar, 2022

Problem 1

a)

$$\begin{aligned}
\mathrm{E}[(y_0 - \hat{f}(x_0))^2] &= \mathrm{E}[(f(x_0) - \hat{f}(x_0) + \varepsilon)^2] \\
&= \mathrm{E}[(f(x_0) - \hat{f}(x_0))^2 + 2\varepsilon(f(x_0) - \hat{f}(x_0)) + \varepsilon^2] \\
&= \mathrm{E}[(f(x_0) - \hat{f}(x_0))^2] + 2\mathrm{E}[\varepsilon]\mathrm{E}[f(x_0) - \hat{f}(x_0)] + \mathrm{E}[\varepsilon^2] \\
&= \mathrm{E}[f(x_0)^2] - 2\mathrm{E}[f(x_0)\hat{f}(x_0)] + \mathrm{E}[\hat{f}(x_0)^2] + \mathrm{Var}[\varepsilon] + \ textE[\varepsilon]^2 \\
&= f(x_0)^2 - 2f(x_0)\mathrm{E}[\hat{f}(x_0)] + \mathrm{Var}[\hat{f}(x_0)] + \mathrm{E}[\hat{f}(x_0)]^2 + \mathrm{Var}[\varepsilon] \\
&= (f(x_0) - \mathrm{E}[\hat{f}(x_0)])^2 + \mathrm{Var}[\hat{f}(x_0)] + \mathrm{Var}[\varepsilon] \\
&= \mathrm{Bias}[\hat{f}(x_0)]^2 + \mathrm{Var}[\hat{f}(x_0)] + \sigma^2
\end{aligned}$$

b)

The first term is the squared bias, which is a measure of how well the model captures the underlying structure of the data. The second term is the variance, which describes the spread in the distribution of the estimated function value, i.e., how much the fitted value will tend to change for new data. The third and final term is the irreducible error, which represents the statistical noise from factors that are not included in our model (and thus out of our control).

c)

TRUE, FALSE, TRUE, FALSE

d)

TRUE, FALSE, TRUE, FALSE

e)

iii)

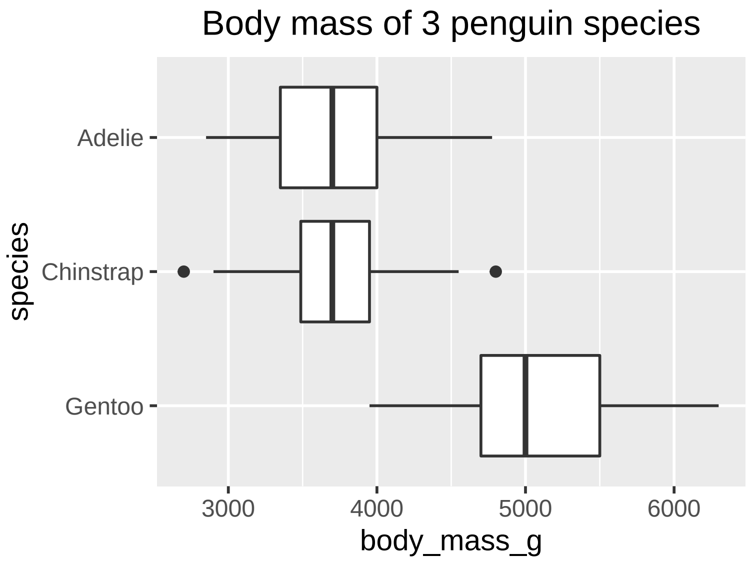
Problem 2

a)

- The first mistake is throwing away the sex variable. The p-value for this covariate is effectively zero, which provides overwhelming evidence against the hypothesis that the true sex coefficient is zero. This makes intuitive sense, because almost anyone could guess that male penguins are significantly heavier than female penguins on average. Basil the cat should definitely include the sex covariate in the model.
- Basil goes on to argue that there is not sufficient evidence that the overall species coefficient is zero, based on the p-values of the individual species coefficients. This is flawed argumentation, because a simple t-test does not solidify that both coefficients might actually zero *simultaneously*. For this, an F-test must be used instead.
- Lastly, even though Basil has already asserted that species is insignificant (which is wrong), he contradicts himself when he uses Chinstrap coefficient to conclude that Chinstrap penguins have the largest body mass, even though the standard error for this coefficient is 771.358, which is a huge proportion of the estimated value of 1008.380. Observing the box plot below makes it clear that this is the wrong conclusion, and that it is in fact the Gentoo penguins which have the largest body mass, according to this data.

b)

```r
library(GGally)
library(palmerpenguins)
data(penguins)

Penguins <- subset(penguins, select = -c(island, year))
ggpairs(Penguins)[1, 5] + ggtitle("Body mass of 3 penguin species") +
  theme(plot.title = element_text(hjust = 0.5))
```



c)

```
library(GGally)
library(palmerpenguins)
data(penguins)

# remove island and year variables
Penguins <- subset(penguins, select = -c(island, year))

# fit model based on expert knowledge
penguin.model <- lm(body_mass_g ~ flipper_length_mm + sex +
                      bill_depth_mm * species,
                      data = Penguins)

# view model coefficients
summary(penguin.model)$coefficients
```

```
##                                   Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)                    -1336.58287 646.922248 -2.066064 3.961450e-02
## flipper_length_mm                 17.37877   2.910449  5.971165 6.172012e-09
## sexmale                          432.90151  44.633685  9.698987 1.059323e-19
## bill_depth_mm                     82.98484  22.324227  3.717255 2.370966e-04
## speciesChinstrap                1460.14721 680.389708  2.146045 3.260954e-02
## speciesGentoo                    644.88114 542.573989  1.188559 2.354811e-01
## bill_depth_mm:speciesChinstrap   -83.53310  37.009147 -2.257093 2.466587e-02
## bill_depth_mm:speciesGentoo       36.17178  34.481962  1.049006 2.949549e-01
```

```
# perform ANOVA to view significance of species interaction terms
anova(penguin.model)
```

```
## Analysis of Variance Table
##
## Response: body_mass_g
##                        Df    Sum Sq   Mean Sq   F value    Pr(>F)
## flipper_length_mm       1 164047703 164047703 1994.7424 < 2.2e-16 ***
## sex                     1   9416589   9416589  114.5013 < 2.2e-16 ***
## bill_depth_mm           1   3667377   3667377   44.5936 1.051e-10 ***
## species                 2  10670525   5335262   64.8743 < 2.2e-16 ***
## bill_depth_mm:species   2    729458    364729    4.4349   0.01258 *
## Residuals             325  26728014     82240
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
# model the body mass as a function of species alone
massVSspecies.model <- lm(body_mass_g ~ species, data = Penguins)
summary(massVSspecies.model)
```

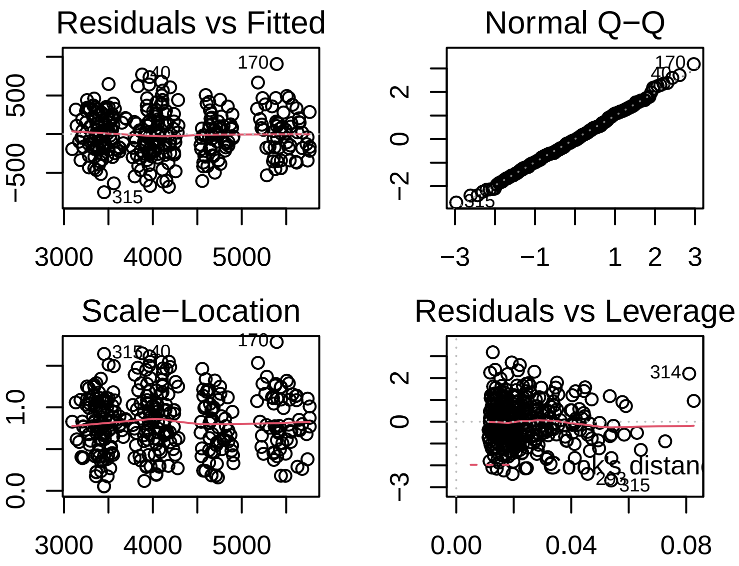
```
##
## Call:
## lm(formula = body_mass_g ~ species, data = Penguins)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1126.02  -333.09   -33.09   316.91  1223.98
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3700.66      37.62   98.37   <2e-16 ***
## speciesChinstrap    32.43      67.51    0.48    0.631
## speciesGentoo     1375.35      56.15   24.50   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 462.3 on 339 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.6697, Adjusted R-squared:  0.6677
## F-statistic: 343.6 on 2 and 339 DF,  p-value: < 2.2e-16
```

```r
# make two other models for comparison

noInteraction.model = basil.model = lm(body_mass_g ~ flipper_length_mm +
                      sex + bill_depth_mm + species,
                      data = Penguins)

basil.model = lm(body_mass_g ~ flipper_length_mm +
                      bill_depth_mm * species,
                      data = Penguins)


# compare adjusted R^2 values for the models
c(summary(penguin.model)$adj.r.squared,
  summary(noInteraction.model)$adj.r.squared,
  summary(basil.model)$adj.r.squared)
```

```
## [1] 0.8731593 0.8704945 0.8334786
```

```r
# make some plots to further evaluate model fit
par(mfrow = c(2,2), mar = c(2,2,2,2))
plot(penguin.model)
```



```r
# only relevant numerical variables of penguin data
pnum <- na.omit(penguins[c(4, 5)])

# empirical covariance and mean of numerical variables
covMat = cov(pnum)
means = sapply(pnum, mean)

# sample size
N <- 20

# create some dummy data
randomSpecies <- sample(c("Adelie", "Chinstrap", "Gentoo"), N, replace = TRUE)
randomBillDepth <- rnorm(N, means[1], sqrt(covMat[1,1]))
randomFlipperLength <- rnorm(N, means[2], sqrt(covMat[2,2]))
randomSex <- sample(c("male", "female"), N, replace = TRUE)
randomAdelie = sample(c(0, 0, 1), N, replace = TRUE)

penguinSample <- data.frame(species = randomSpecies,
                   bill_depth_mm = randomBillDepth,
                   flipper_length_mm = randomFlipperLength,
                   sex = randomSex,
                   adelie = randomAdelie)

# predict body mass of N new penguins
predictions = predict(penguin.model, penguinSample, interval = "confidence")

predictions
```

```
##          fit      lwr      upr
## 1  3632.056 3509.217 3754.896
## 2  5340.683 5011.134 5670.231
## 3  4123.125 3844.305 4401.946
## 4  3625.006 3385.125 3864.887
## 5  4777.523 4689.838 4865.209
## 6  5321.671 5032.760 5610.583
## 7  3887.295 3764.485 4010.106
## 8  3990.334 3804.826 4175.841
## 9  5370.940 5103.263 5638.617
## 10 4927.033 4621.962 5232.104
## 11 3584.913 3467.668 3702.157
## 12 3790.992 3626.179 3955.806
## 13 3434.181 3084.591 3783.771
## 14 5010.661 4836.275 5185.046
## 15 3858.162 3634.978 4081.346
## 16 3912.089 3711.272 4112.906
## 17 3739.627 3575.110 3904.144
## 18 4625.964 4275.293 4976.634
## 19 5260.044 4881.618 5638.470
## 20 4164.054 3953.419 4374.690
```

REPORT: PREDICTION OF PENGUIN BODY MASS, by group 4, the humans :)

We start with fitting a model in the same manner as Basil, since it is based on expert knowledge about the penguins. It is evident that flipper length, sex, and bill depth are all significant (sex is a binary factor, so it is sufficient to consider the p-value directly). The species factor and its interaction with bill depth has more than two levels: three, to be precise. Therefore, we perform ANOVA to test the null hypotheses that the species coefficients, as well as the interaction coefficients, are simultaneously zero. The p-values in the ANOVA table provide sufficient evidence to reject these hypotheses, so the species and interaction between species and bill depth overall are both significant, and should be included in our model. Now that we have determined species as an important factor, we make a smaller model with a species variable only, to compare the individual species. It is evident that the Gentoo penguins tend to be significantly more massive than both the Adelies and the Chinstraps, and that there is no discernable difference between Adelies and Chinstraps. This can also be observed graphically from the plot in **b)**.

Next, we evaluate the goodness-of-fit by comparing the adjusted R^2-values for three different models. The first one is the one we went with (including sex and interactions), the second is one including sex but no interactions, and the third one is Basil's final model. The value for our chosen model suggests a better fit than the two others, even though the adjusted R^2-value penalizes for adding including covariates.

Additionally, we have added four plots to evaluate our model. The Residuals vs Fitted and Scale-Location plots look good; the residuals are roughly evenly spread around zero, and there is no trend, which indicates zero expected value and homoscedasticity of the errors. The QQ plot looks nice and straight, which suggests normally distributed errors. Also, the leverage plot reveals that there are not any large outliers in our data relative to the fitted model.

Lastly, we thought we would actually create some dummy data to obtain predictions. We based it on empirical variances and means of the continuous variables. Of course, we realize that the explanatory variables are not uncorrelated, but they were nevertheless sampled independently for simplicity (continuous variables from normal distributions and categorical variables from uniform distributions).

Remarks: If Basil's goal was to merely determine which type of penguin has the largest body mass, he could have created a much simpler model in order to infer this (for example like massVSspecies.model above). Even simpler, just plot and observe the data itself. Judging from the problem description, it seems like Bssil was tasked to create a model to *make predictions* about a penguin's body mass, given sex, species, flipper length and bill depth. He has completely failed to do this, and should be ashamed of himself. But after all, he is a cat, and perhaps the most criminal mistake of all was to to allow him to conduct the analysis to begin with.

Problem 3

```
#install.packages("palmerpenguins")  # Run if you haven't installed this before.
library(palmerpenguins) # Contains the data set "penguins".
data(penguins)
# Remove island, and year variable, as we won't use those.
Penguins <- subset(penguins, select = -c(island, year))

library(tidyverse)
#library(GGally)
# Create a new boolean variable indicating whether or not the penguin is an Adelie penguin
Penguins$adelie <- ifelse(Penguins$species == "Adelie", 1, 0)
# Select only relevant variables and remove all rows with missing values in
# body mass, flipper length, sex or species.
Penguins_reduced <- Penguins %>%
  dplyr::select(body_mass_g, flipper_length_mm, adelie) %>%
  mutate(body_mass_g = as.numeric(body_mass_g),
         flipper_length_mm = as.numeric(flipper_length_mm)) %>%
  drop_na()
set.seed(4268)
# 70% of the sample size for training set
training_set_size <- floor(0.70 * nrow(Penguins_reduced))
train_ind <- sample(seq_len(nrow(Penguins_reduced)), size = training_set_size)
train <- Penguins_reduced[train_ind, ]
test <- Penguins_reduced[-train_ind, ]
```

a)

```
set.seed(4268)

# i) Logistic regression
log.fit<-glm(adelie ~ body_mass_g + flipper_length_mm, family="binomial", data=train)

log.fit.probs=predict(log.fit, newdata=test, type="response")

log.fit.preds=ifelse(log.fit.probs>0.5, 1, 0)

log.conf.mat=table(test$adelie, log.fit.preds)

# ii) QDA
qda.fit<-qda(adelie ~ body_mass_g + flipper_length_mm, data=train)

qda.fit.prob=predict(qda.fit, newdata=test)$posterior
qda.fit.pred=predict(qda.fit, newdata=test)$class

qda.conf.mat=table(test$adelie, qda.fit.pred)

# iii) KNN
knn.train=as.matrix(train)
knn.test=as.matrix(test)
```

```r
knn.fit=knn(train=knn.train, test=knn.test, cl=train$adelie, k=25, prob=T)

knn.conf.mat=table(test$adelie, knn.fit)

#iv) Sensitivity and specificity
sens.and.spec<-function(table){
  TP=table[2, 2]
  P=table[2, 1]+table[2, 2]
  TN=table[1, 1]
  N=table[1, 1]+table[1, 2]
  sens=TP/P
  spec=TN/N
  return (c(sens, spec))
}

#LogReg fit:
log.results=sens.and.spec(log.conf.mat)
log.sens=log.results[1]
log.spec=log.results[2]

#QDA fit:
qda.results=sens.and.spec(qda.conf.mat)
qda.sens=qda.results[1]
qda.spec=qda.results[2]

#KNN fit:
knn.results=sens.and.spec(knn.conf.mat)
knn.sens=knn.results[1]
knn.spec=knn.results[2]
```

```r
cat("Sensitivity for logistic regression: ", log.sens, "\nSpecificity for logistic regression:", log.spe
```

```
## Sensitivity for logistic regression:  0.9767442
## Specificity for logistic regression: 0.8666667
```

```r
cat("Sensitivity for QDA: ", qda.sens, "\nSpecificity for QDA:", qda.spec)
```

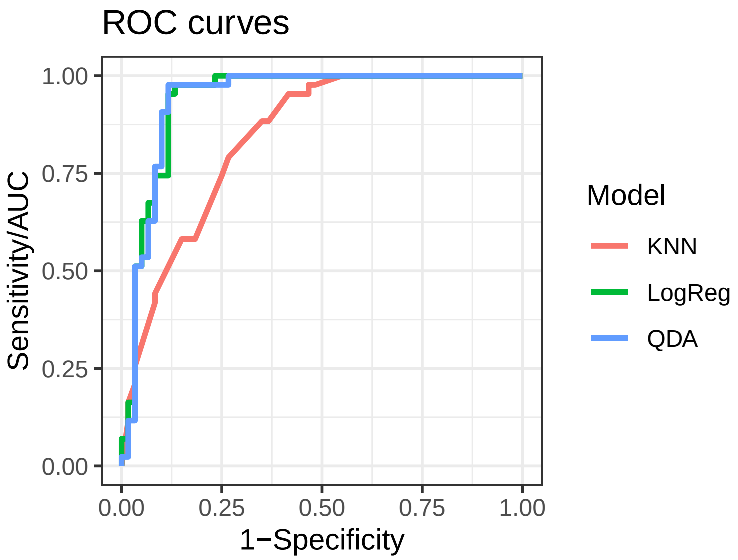
```
## Sensitivity for QDA:  0.9767442
## Specificity for QDA: 0.7666667
```

```r
cat("Sensitivity for KNN: ", knn.sens, "\nSpecificity for KNN:", knn.spec)
```

```
## Sensitivity for KNN:  0.9534884
## Specificity for KNN: 0.5833333
```

b)

```r
set.seed(4268)
# i) ROC plots

logRoc=roc(response=test$adelie, predictor=log.fit.probs, direction="<")

AUClogreg=auc(logRoc)

qdaRoc=roc(response=test$adelie, predictor=qda.fit.prob[, 2], direction="<")

AUCqda=auc(qdaRoc)

knn.probs=attributes(knn.fit)$prob
not.Adelie=which(knn.fit=="0")

knn.probs[not.Adelie]=1-knn.probs[not.Adelie]

knnRoc=roc(response=test$adelie, predictor=knn.probs, direction="<")

AUCknn=auc(knnRoc)

dat=data.frame(Adelie=test$adelie, LogReg=log.fit.probs, QDA=qda.fit.prob[, 2], KNN=knn.probs)

dat_long=melt_roc(dat, "Adelie", c("LogReg", "QDA", "KNN"))
ggplot(dat_long, aes(d=D, m=M, color=name)) +
  geom_roc(n.cuts=F) + xlab("1-Specificity") +
  ylab("Sensitivity/AUC") +
  labs(title="ROC curves") +
  scale_color_discrete(name="Model") +
  theme_bw()
```



```r
cat("AUC for the logstic regression: ", AUClogreg, "\nAUC for the QDA: ", AUCqda, "\nAUC for the KNN: "
```

```
## AUC for the logstic regression:  0.9391473
## AUC for the QDA:  0.9379845
## AUC for the KNN:  0.8416667
```

ii) The ROC curve plots the true positive rate aginst the false positive rate as the threshold, the value that decide how the model classifies, varies. An optimal ROC curve is a curve such that both the sensitivity and specificity are as close to one as possible. In the plot this would mean that the graph is close to the top left corner, meaning that both the sensitivity and specificity are high. Furthermore, the AUC calculates the area under the ROC curve, so the optimal AUC is equal to 1. Based on this, we can see from the ROC plot that the logistic regression model and the QDA model generally gives a better balance between sensitivity and specificity and is far more accurate in both measures, than the KNN. We conclude that KNN performs worse than the other two. To seperate the logistic regression and the QDA, we notice that the AUC of the logistic regression is slightly higher than that of the QDA. The AUC alone might not be enough to evaluate performance, but at our threshold (0.5), the specificity of the logistic regression is significantly higher than that of the QDA. This implies that the logistic regression performs the best out of our three models. However, one should be careful as this is not the case for all threshold values, and for some threshold values the QDA might give more accurate predictions.

iii) We feel that the concept of the KNN model is simple, and very easy to grasp. We have used $K = 25$, so for an observed animal we search the neighbourhood of its data and if 13 or more of the neihgbours are Adelie, we classify it as Adelie. However, the KNN model does not provide any information on the effects of our two covariates. Since the KNN is non-paramteric, we only really get information about the output, whereas the parametric methods can give inference about the covariates. Separating the logistic regression and the QDA, we note that the log-odds of the logistic regression can be written as a linear function in the covariates. This simple method might be to prefer over the more complex QDA. For interpretation purposes, one might want a higher bias than variance, as we then know that the errors mostly stems from model simplifications. In our case, with two classes, we feel that less is more and we stick to the simple yet efficient logistic regression.

c)

We are to find the odds ratio that an abserved animal is from the Adelie species, given that we increase the body mass by $1000g$. The odds of an event is given by

$$odds(Y_i = 1|X_j = x_{ij}) = \frac{p_i}{1-p_i} = \frac{P(Y_i = 1|X = x)}{P(Y_i = 0|X = x)} = e^{\beta_0+\beta_1 x_{i1}+...+\beta_p x_{ip}}$$

In this case, we only have two covariates, body mass(β_1) and flipper length(β_2). This means our expression for the odds ratio becomes

$$\frac{odds(Y_i = 1|X_j = x_{ij} + 1000)}{odds(Y_i = 1|X_j = x_{ij})} = \frac{e^{\beta_0+\beta_1(x_{i1}+1000)+\beta_2 x_{i2}}}{e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}}} = e^{1000\beta_1}$$

Our coefficients can easily be read of the summary of our logistic regression model, and we obtain $\beta_1 = 0.000712$. This means that if the body mass were to increase by 1000g, the odds that an observed animal is from the Adelie species would be multiplied by $e^{1000\beta_1} = 2.038$.

Alternative *iii*) is correct.

d)

```r
set.seed(4268)
library(ggpubr)
```

```
log.fit.test<-glm(adelie ~ body_mass_g + flipper_length_mm, family="binomial", data=train)

log.fit.test.probs=predict(log.fit, newdata=Penguins_reduced, type="response")

log.fit.test.preds=ifelse(log.fit.test.probs>0.5, 1, 0)

Penguins_reduced$log.fit.test.preds=log.fit.test.preds

Penguins_reduced$errors=abs(Penguins_reduced$adelie - Penguins_reduced$log.fit.test.preds)

true.plot<-ggplot(data=Penguins_reduced)+
  geom_point(mapping=aes(x=body_mass_g, y=flipper_length_mm, color=as.factor(adelie), shape=as.factor(e
  scale_color_manual(values=c('green', 'blue'), labels=c('Not Adelie', 'Adelie'), name=c('Species')) +
  scale_shape_discrete(name='Predictions', labels=c('Correct', 'Wrong')) +
  ggtitle(label="True and predicted values") +
  xlab("Body mass [g]") +
  ylab("Flipper length [mm]") +
  theme_bw()

true.plot
```

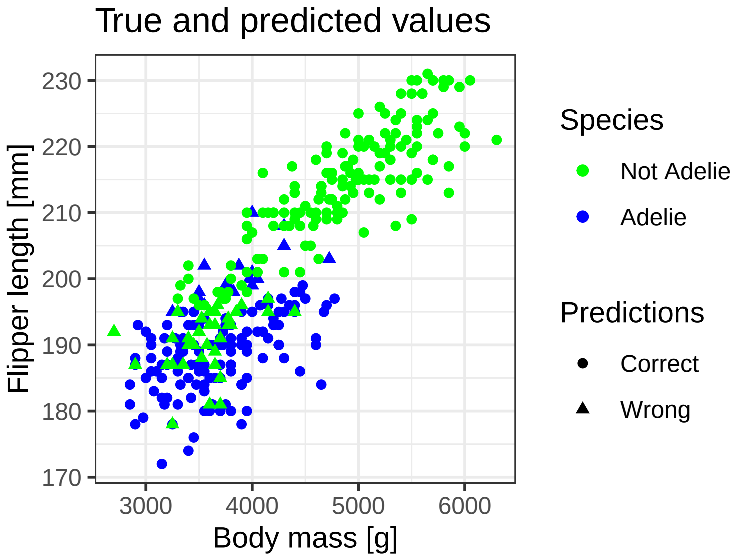


Figure 1: The blue color indicates an Adelie, the green a different species. The circle shapes indicates a correct predicition from our model, while the traingles indicate that our model predicted the animal wrongly, compared to the actual data set.

Problem 4

a)

i) True
ii) False

iii) False
iv) False

```
### 4b)
### Load data
id <- "1chRpybM5cJn4Eow3-_xwDKPKyddL9M2N"  # google file ID
d.chd <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download", id))

##Make the Logistic regression
model <- glm( chd ~ sex + sbp + smoking, data = d.chd, family = binomial)

#Make an array of the estimated beta values
betas <- c(summary(model)$coef[1], summary(model)$coef[2], summary(model)$coef[3], summary(model)$coef[4

#Takes in the beta-array and and array of the x-values (covariates) we want to
#evaluate at
probability_func <- function(betas) {
  return(1/(1+ exp(-(betas[1] + betas[2]*1 + betas[3]*150  +
                      betas[4]*0 ))))
}

#The estimated probability
estimated_prob = probability_func(betas)
betas
```

```
## [1] -6.65883685 -1.34351384  0.03877165  0.41031080
```

```
estimated_prob
```

```
## [1] 0.10096
```

We define X_1= sex, X_2= systolic blood preasure (sbp), and X_3= smoking, and we estimate the $\beta_i, i = 0, .., 3$ in R by using the glm-method. The result we obtain is

$$\vec{\beta} = [\beta_0, \beta_1, \beta_2, \beta_3] = [-6.65884, -1.34351, 0.03877, 0.41031],$$

We then plug the values into the probability function (using the shorthand CHD = Coronary heart disease),

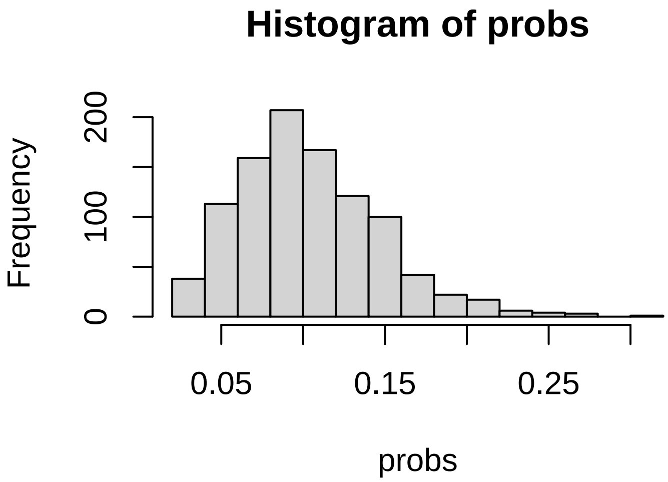
$$\Pr\{\text{CHD}|x_1, x_2, x_3\} = \frac{1}{1 + \exp\Bigg(- (\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3)\Bigg)}$$

and by plugging in $(x_1, x_2, x_3) = (1, 150, 0)$, we get

$$\hat{p} = \Pr\{\text{CHD}|1, 150, 0\} \simeq 0.10.$$

```
#4 c)
#Sample size
n = 500
B<- 1000
probs <- matrix(0, 1, B) #Vector of probabilities
```

```r
set.seed(4268)

# Bootstrapping process
for (val in 1: B)
{
  boot <- d.chd[sample(n, n, replace = TRUE),]
  bootmodel <- glm( chd ~ sex + sbp + smoking, data = boot, family = binomial)
  bootbetas <- c(summary(bootmodel)$coef[1], summary(bootmodel)$coef[2], summary(bootmodel)$coef[3],  su
  probs[val] = probability_func(bootbetas)
}
hist(probs)
```



```r
mean_prob = mean(probs)
mean_prob
```

```
## [1] 0.1040204
```

```r
#Standard Error
std_error = sd(probs)
std_error
```

```
## [1] 0.04301888
```

```r
quantile(probs, probs = c(0.025, 0.975))
```

```
##       2.5%      97.5%
## 0.03791544 0.20520844
```

We now wish to use the bootstrapping method to estimate the uncertainty in the probability we found in b). We set $B = 1000$ and create a for-loop where we in each iteration draw sample of size $n = 500$ with replacement from our data set. We then do the same as we did in b) by performing the logistic regression from that sample, and plugging in the $\beta_i, i = 0, .., 3$ in the probability function. For each iteration b, we insert the probability p_b into an array $\vec{p} = [p_1, ..., p_b, ..., p_B]$. We easily find the average probability $\bar{p} = \frac{1}{B}\sum_{b=1}^{B} p_b = 0.107722$. We find the standard error of the bootstrap estimate by the formula

$$SE_B(\hat{p}) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(p_b - \bar{p})} = 0.0430.$$

Finally, we find the 95% confidence interval by inspecting the quantiles in our probability sample $\vec{p}$. We use the *quantile* function in R to find the 2.5% and the 97.5% quantiles, and the result is

$$\text{CI} = [0.03791544, 0.20520844]$$

We see that the CI is quite large, and range from around 4% to 21%. We further see that the average of the bootstrapping values $\bar{p}$ is very close to our first estimation $\hat{p}$. Given the standard error of around 4%, which makes up about 40% of the mean, it is hard to estimate an accurate probability of coronary heart disease for a non-smoking male with a systolic blood pressure of 150. The plausible values lies indeed within this interval, and it is likely with 95% probability that the true value lies in this interval.

```r
### 4d)
summary(glm( chd ~ sex + sbp + smoking, data = d.chd, family = binomial))$coef
```

```
##                Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -6.65883685 2.36740155 -2.812720 4.912446e-03
## sex         -1.34351384 0.32148322 -4.179110 2.926516e-05
## sbp          0.03877165 0.01793731  2.161508 3.065610e-02
## smoking      0.41031080 0.31014166  1.322979 1.858425e-01
```

i) False.
ii) False.
iii) True.
iv) True.