Compulsory Exercise 1 – Solution

TMA4268 Statistical Learning V2022

Emma Skarstein, Daesoo Lee, Stefanie Muff, Department of Mathematical Sciences, NTNU

Hand out date: February 7, 2022

Problem 1 (8P)

We have a univariate continuous random variable Y and a covariate x. Further, we have observed a training set of independent observation pairs $\{x_i, y_i\}$ for $i = 1, \ldots, n$. Assume a regression model

$$Y_i = f(x_i) + \varepsilon_i \ ,$$

where f is the true regression function, and ε_i is an unobserved random variable with mean zero and constant variance σ^2 (not dependent on the covariate). Using the training set we can find an estimate of the regression function f, and we denote this by $\hat{f}$. We want to use $\hat{f}$ to make a prediction for a new observation (not dependent on the observations in the training set) at a covariate value x_0. The predicted response value is then $\hat{f}(x_0)$. We are interested in the error associated with this prediction.

a) (2P)

Derive the decomposition of the expected test MSE, $E[y_0 - \hat{f}(x_0)]^2$, into three terms (bias, variance, and irreducible error).

Solution:

Useful rule: $\text{Var}[Y] = \text{E}[Y^2] - \text{E}[Y]^2$.

Use that $y_0 = f(x_0) + \varepsilon$:

$$\begin{aligned}
\text{E}[y_0 - \hat{f}(x_0)]^2 &= \text{E}[f(x_0) + \varepsilon - \hat{f}(x_0)]^2 \\
&= \text{E}[f(x_0)^2] + \text{E}[\varepsilon^2] + \text{E}[\hat{f}(x_0)^2] - 2\text{E}[f(x_0)\hat{f}(x_0)] - \underbrace{2\text{E}[\varepsilon\hat{f}(x_0)]}_{=0} + \underbrace{2\text{E}[\varepsilon f(x_0)]}_{=0} \ ,
\end{aligned}$$

where the last two terms are zero due to the independence of ϵ with anything else, and $\text{E}[\varepsilon] = 0$. Using that $\text{E}[\hat{f}(x_0)^2] = \text{Var}[\hat{f}(x_0)^2] + \text{E}[\hat{f}(x_0)]^2$, $\text{E}[f(x_0)^2] = f(x_0)^2$ and $\text{E}[\varepsilon^2] = \text{Var}[\varepsilon]$, we get

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

where the last equation just rearranges the terms and uses that $[\text{Bias}(\hat{f}(x_0))]^2 = (f(x_0) - \text{E}[\hat{f}(x_0)])^2$.

b) (1P)

Explain with words how we can interpret the three terms.

Solution:

$$\mathrm{E}[(Y - \hat{f}(x_0))^2] = \mathrm{Var}(\varepsilon) + \mathrm{Var}[\hat{f}(x_0)] + [\mathrm{Bias}(\hat{f}(x_0))]^2$$

- First term: irreducible error, σ^2 and is always present unless we have measurements without error. This term cannot be reduced regardless how well our statistical model fits the data.
- Second term: variance of the prediction at x_0 or the expected deviation around the mean at x_0. If the variance is high, there is large uncertainty associated with the prediction.
- Third term: squared bias. The bias gives an estimate of how much the prediction differs from the true mean $f(x_0)$. If the bias is low the model gives a prediction which is close to the true value.

c) (2P) - Multiple choice

Figure 1 shows the squared bias, variance, irreducible error and total error for increasing values of K in KNN regression. Which of the following statements are true and which are false? Say for *each* of them if it is true or false.

(i) Decreased K corresponds to increased flexibility of the model.
(ii) The variance increases with increased value of K.
(iii) The blue line corresponds to the irreducible error.
(iv) The squared bias decreases with increased value of K.

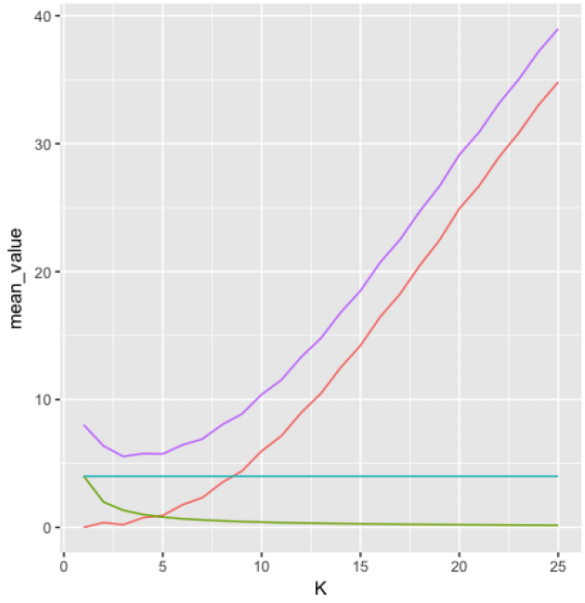


Figure 1: Squared bias, variance, irreducible error and total error for increasing values of K in KNN

Solution:

TRUE, FALSE, TRUE, FALSE
Explanation:

(ii) TRUE: Decreasing K means that we include less points when classifying, so decreased K corresponds to increased flexibility of the model.

(iii) FALSE: The green line corresponds to the variance, which decreases for increased K.
(iv) TRUE: Blue line is the irreducible error, constant for all K
(v) FALSE: Orange curve corresponds to squared bias.

d) (2P) Multiple choice

Which of the following statements are true and which are false? Say for *each* of them if it is true or false.

(i) If the relationship between the predictors and response is highly non-linear, a flexible method will generally perform better than an inflexible method.
(ii) If the number of predictors p is extremely large and the number of observations n is small, a flexible method will generally perform better than an inflexible method.
(iii) In KNN classification, it is important to use the test set to select the value K, and not the training set, to avoid overfitting.
(iv) In a linear regression setting, adding more covariates will reduce the variance of the predictor function.

Solution: TRUE-FALSE-FALSE-FALSE

(i) A flexible method will be better able to identify the highly non-linear relationship.
(ii) A flexible learning method would perform worse because it would be more likely to overfit.
(iii) A validation set should be used, the test set should just be used to evaluate the final model.
(iv) More covariates will increase variance.

e) (1P) Single choice

$\mathbf{X} = [x_1, x_2, x_3]^T$ is a 3-dimensional random vector with covariance matrix

$$\mathbf{\Sigma} = \begin{bmatrix} 50 & 33 & 18 \\ 33 & 38 & -10 \\ 18 & -10 & 72 \end{bmatrix}$$

The correlation between element x_1 and x_2 of the vector $\mathbf{X}$ is:

(i) 0.017
(ii) -0.19
(iii) 0.76
(iv) 0.66
(v) 0.10
(vi) 0.3
(vii) It is not possible to calculate the correlation, because this is not a proper covariance matrix.

Solution: (iii): $\frac{33}{\sqrt{50 \cdot 38}} = 0.76$

Problem 2 (9P)

In the following example, Basil has been given a dataset by his boss. The dataset consists of observations of Antarctic penguins who live on the Palmer Archipelago. Basil's boss wonders if he can set up a model to predict the body mass of a given penguin based on some recorded characteristics of the penguin, which are specified in advance based on expert knowledge. However, Basil is a cat, and despite being very clever, he has only a very rudimentary knowledge of statistical techniques and data analysis. In the following code and report, Basil has made a couple of very problematic mistakes.

```r
############## =^._.^= ~~~BASIL'S CODE~~~ =^._.^= ##############
############## 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))

# Fit the model as specified in advance based on expert knowledge:
penguin.model <- lm(body_mass_g ~ flipper_length_mm + sex + bill_depth_mm * species,
    data = Penguins)

# Look at the 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
```

```r
# Fit final model without sex
final.model <- lm(body_mass_g ~ flipper_length_mm + bill_depth_mm * species, data = Penguins)

summary(final.model)
```

```
##
## Call:
## lm(formula = body_mass_g ~ flipper_length_mm + bill_depth_mm *
##     species, data = Penguins)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -895.42 -226.28  -24.56  207.65 1074.66
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -4213.176    647.731  -6.505 2.84e-10 ***
## flipper_length_mm                  24.621      3.173   7.760 1.04e-13 ***
## bill_depth_mm                     176.443     22.580   7.814 7.22e-14 ***
## speciesChinstrap                 1008.380    771.358   1.307   0.1920
## speciesGentoo                     129.453    608.383   0.213   0.8316
## bill_depth_mm:speciesChinstrap    -61.538     41.978  -1.466   0.1436
## bill_depth_mm:speciesGentoo        78.026     38.545   2.024   0.0437 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 327.3 on 335 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.8364, Adjusted R-squared:  0.8335
```

```
## F-statistic: 285.5 on 6 and 335 DF,  p-value: < 2.2e-16
```

REPORT: PREDICTION OF PENGUIN BODY MASS, by Basil Thecat :3

We begin with a linear regression model with body mass as the response, and flipper length, bill depth, species, and sex as covariates, as well as an interaction effect between bill depth and species. In this first model, the sex covariate has the smallest p-value and is thus excluded in the final model to avoid overfitting. The final model can be described depending on the species of the penguin:

$$\hat{y}_{adelie} = \hat{\beta}_0 + \hat{\beta}_{flipper\_length} x_{flipper\_length} + \hat{\beta}_{bill\_depth} x_{bill\_depth}$$
$$\hat{y}_{chinstrap} = \hat{\beta}_0 + \hat{\beta}_{flipper\_length} x_{flipper\_length} + (\hat{\beta}_{bill\_depth} + \hat{\beta}_{bill\_depth:chinstrap}) x_{bill\_depth} + \hat{\beta}_{chinstrap}$$
$$\hat{y}_{gentoo} = \hat{\beta}_0 + \hat{\beta}_{flipper\_length} x_{flipper\_length} + (\hat{\beta}_{bill\_depth} + \hat{\beta}_{bill\_depth:gentoo}) x_{bill\_depth} + \hat{\beta}_{gentoo}$$

(where $\hat{y}_{adelie}$ is the predicted body mass for Adelie penguins, $\hat{\beta}_0$ is the estimated intercept, $x_{flipper\_length}$ is the flipper length covariate, $\hat{\beta}_{flipper\_length}$ is the estimated flipper length coefficient, etc.) Since both of the species coefficients have large p-values, we do not reject the null hypothesis that the species coefficient overall is actually zero. For the interaction effect between species and bill depth, the Gentoo interaction is significant ($p < 0.05$), so the interaction term overall is significant. Based on the coefficient for the dummy variable for the chinstrap penguins being the largest ($\hat{\beta}_{chinstrap} \approx 1008$), we can tell that the chinstrap penguins have the largest body mass.

a) (3P)

Identify three of the mistakes Basil made (there are more than three, but only report three). List them as bullet points along with brief explanations of why these are inappropriate modeling choices.

Solution:

Mistakes:

- "In this first model, the sex covariate has the smallest p-value and is thus excluded in the final model to avoid overfitting". Shouldn't have removed the sex-covariate.
- "Since both of the species coefficients have large p-values, we do not reject the null hypothesis that the species coefficient overall is actually zero." Need anova test to conclude anything about the overall covariate. Also, the main effects cannot really be interpreted in the presence of an interaction term (so here are two possible mistakes the students can find, althought he latter has not really been discussed in the course).
- "For the interaction effect between species and bill depth, the Gentoo interaction is significant, so the interaction term overall is significant." Here there are two problems (mentioning each of them is 1P):
  (i) First, we need the F-teest from the `anova()` table to conclude anything about the overall covariate (1P for this one).
  (ii) Second (and more subtle, so we don't expect the students to find this one, but they will get a point if they do), the $p < 0.05$ threshold and significance language should be avoided.
- "Based on the coefficient for the dummy variable for the chinstrap penguins being the largest ($\hat{\beta}_{chinstrap} \approx 1008$), we can tell that the chinstrap penguins have the largest body mass." No, to maket his claim, we should directly look at the data. It turns out that the Gentoo penguins actually have the largest bodymass. [Why is this? In the presence of interaction terms, we cannot draw any conclusions by only looking at the main effect.]

b) (1P)

In order to make an improved model, you will need to understand the data. Create at least one informative plot that helps you explain at least one of Basil's mistakes and that will justify your modeling choices in the

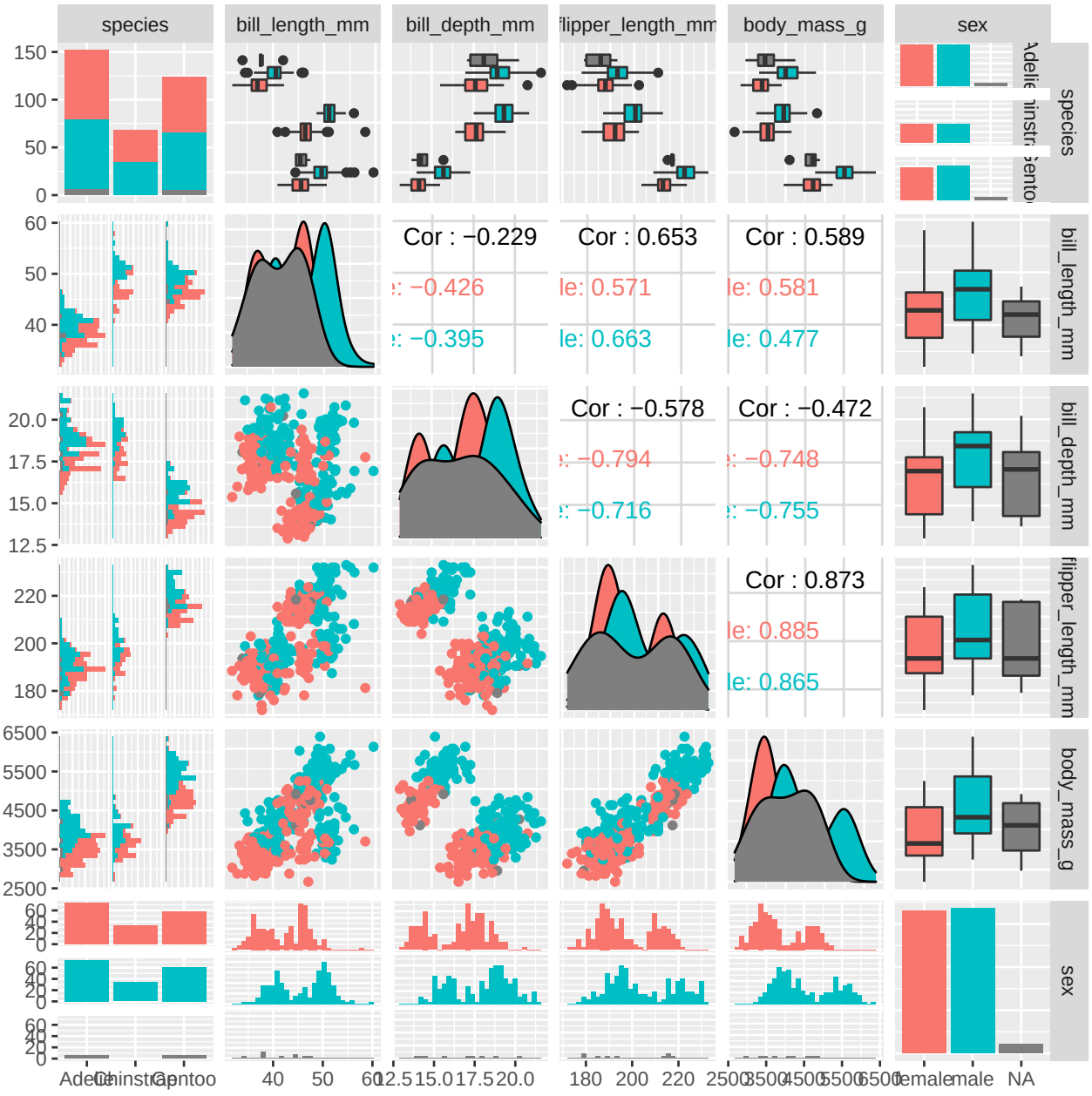
next step. (Plotting the data may even help you *discover* Basil's mistakes in the first place.)

Solution:

There are several ways the students can make plots. Two ideas are presented here (but the code does not need to match, it is ok to generate plots using base R).

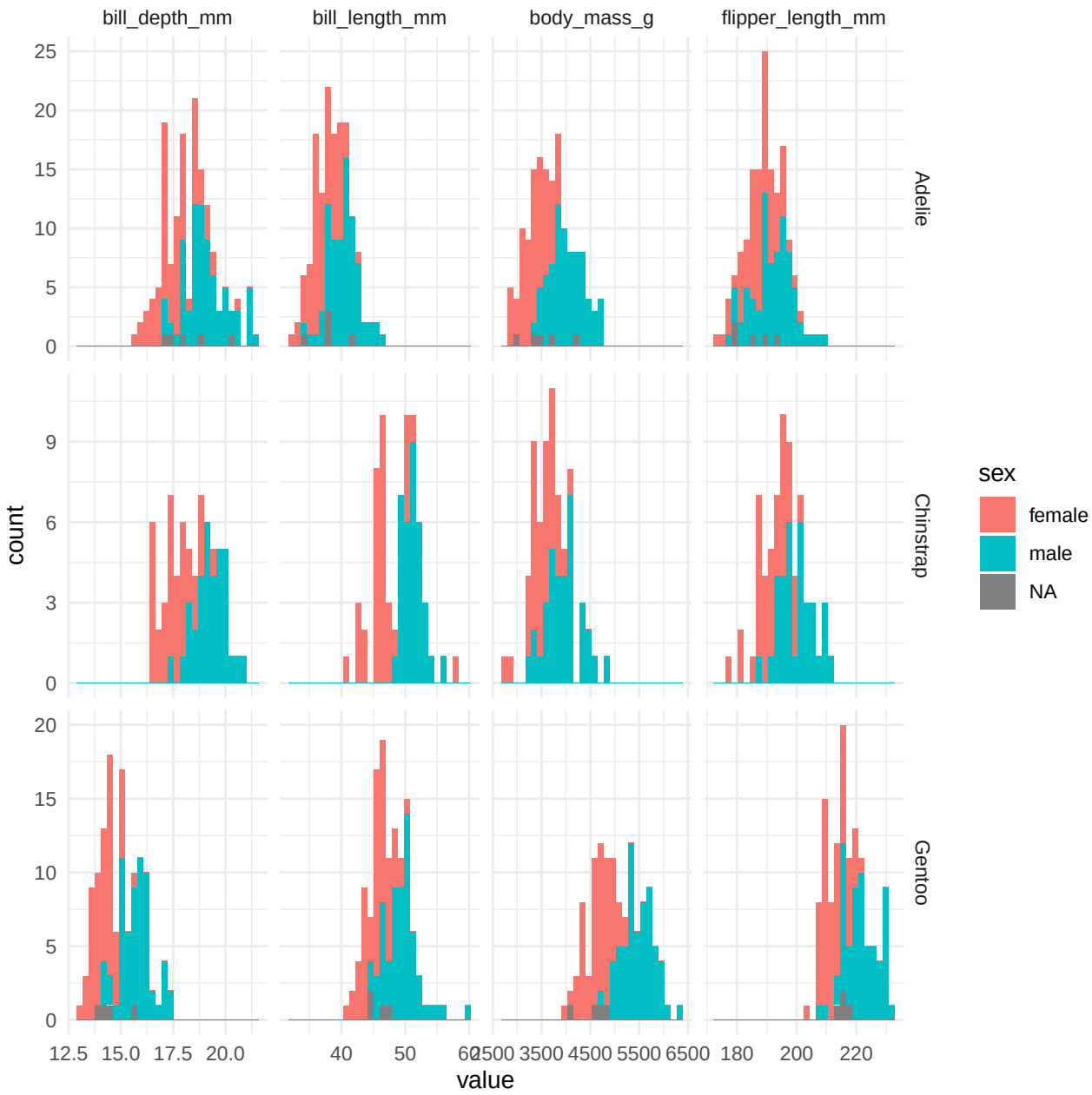
```
library(tidyverse)
library(GGally)

ggpairs(Penguins, aes(color = sex))
```




```
penguins.long <- Penguins %>% dplyr::select(species, bill_length_mm, bill_depth_mm,
    flipper_length_mm, body_mass_g, sex) %>% pivot_longer(penguins, cols = c(bill_length_mm,
    bill_depth_mm, flipper_length_mm, body_mass_g), names_to = "variable")
```

```
ggplot(penguins.long) + geom_histogram(aes(x = value, fill = sex)) + facet_grid(cols = vars(variable),
    rows = vars(species), scales = "free") + theme_minimal()
```



c) (5P)

Redo Basil's analysis including code and report, this time doing it right (4P). Evaluate the fit of the model with at least one graphical tool (1P).

Solution:

For each mistake in the analysis and the report we deduct either -0.5P (for small mistakes) or -1P.

```r
############## =^._.^= ~~~BASIL'S CORRECTED CODE~~~ =^._.^= ##############
############## install.packages('palmerpenguins') # Run if you haven't installed this before.
library(palmerpenguins)  # Contains the data set 'penguins'.

# Remove island and year variable, as we won't use those.
Penguins <- subset(penguins, select = -c(island, year))

# Fit the model
penguin.model <- lm(body_mass_g ~ flipper_length_mm + sex + species * bill_depth_mm,
    data = drop_na(Penguins))  # The drop_na() function here is not necessary, as lm removes the obs wi

# Look at the model coefficients
summary(penguin.model)
```

```
##
## Call:
## lm(formula = body_mass_g ~ flipper_length_mm + sex + species *
##     bill_depth_mm, data = drop_na(Penguins))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -751.2 -183.8   -9.8  191.1  906.9
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -1336.58     646.92  -2.066 0.039615 *
## flipper_length_mm               17.38       2.91   5.971 6.17e-09 ***
## sexmale                        432.90      44.63   9.699  < 2e-16 ***
## speciesChinstrap              1460.15     680.39   2.146 0.032610 *
## speciesGentoo                  644.88     542.57   1.189 0.235481
## bill_depth_mm                   82.98      22.32   3.717 0.000237 ***
## speciesChinstrap:bill_depth_mm -83.53      37.01  -2.257 0.024666 *
## speciesGentoo:bill_depth_mm     36.17      34.48   1.049 0.294955
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 286.8 on 325 degrees of freedom
## Multiple R-squared:  0.8758, Adjusted R-squared:  0.8732
## F-statistic: 327.5 on 7 and 325 DF,  p-value: < 2.2e-16
```

```r
# Look at anova output to check the overall coefficients
anova(penguin.model, test = "Chisq")
```

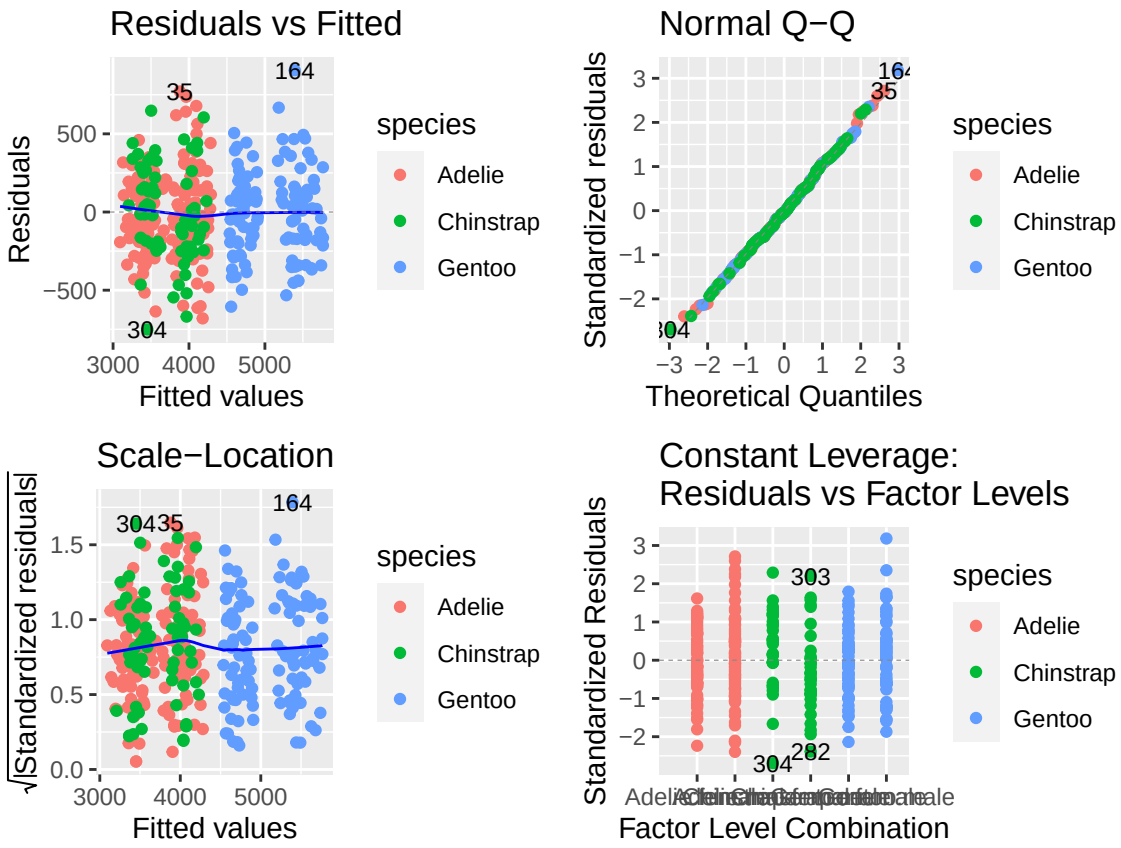
```
## Analysis of Variance Table
##
## Response: body_mass_g
##                         Df    Sum Sq   Mean Sq   F value    Pr(>F)
## flipper_length_mm        1 164047703 164047703 1994.7424 < 2.2e-16 ***
## sex                      1   9416589   9416589  114.5013 < 2.2e-16 ***
## species                  2  13141806   6570903   79.8991 < 2.2e-16 ***
## bill_depth_mm            1   1196096   1196096   14.5440 0.0001638 ***
## species:bill_depth_mm    2    729458    364729    4.4349 0.0125820 *
## Residuals              325  26728014     82240
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In the rewritten report, they need to have corrected all the mistakes from above, as well as added the sex-covariate to the model formulas. They also need to have a plot to evaluate the model fit, for example

```
library(ggfortify)

autoplot(penguin.model, ncol = 2, label.size = 3, data = drop_na(Penguins), colour = "species")
```



The modeling assumptions seem to be fulfilled (0.5P for the graph, 0.5P for the correct conclusion). We do not expect a colored figure.

Problem 3 (13P)

We will now consider the Palmer penguin dataset again, but this time looking at classifying the species of the penguins for a given body mass and flipper length. Since there are three penguin species, for simplicity we will define the goal to be to classify a penguin as belonging to the species Adelie or *not* Adelie, giving us a two-class classification problem instead of three.

The following code modifies the data set for this simplified setting, converts the variables to numeric (because the `knn` function can't handle the `int` class, and will give an error), and removes any missing observations. Please remember to use the same seed when you split the data into training and test set.

```
# 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
```

```r
# 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(0.7 * 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) (5P)

(i) (1P) Fit a **logistic regression** model using the training set, and perform the classification on the test set, using a 0.5 cutoff.

(ii) (1P) Fit a **QDA** model using the training set, and perform the classification on the test set, using a 0.5 cutoff.

(iii) (1P) Finally, do the same as in (i) and (ii) using **KNN** with $k = 25$ (use the `knn` function from the `class` package).

R-hints: In the `knn()` function set `prob=T` to ensure you get the class probabilities that you then need in d):

```r
knnMod = knn(train = ..., test = ..., cl = ..., k = 25, prob = T)
```

(iv) (2P) Calculate the sensitivity and specificity for the three predictions performed on the test set in (i) - (iii).

Solution:

```r
formula <- adelie ~ flipper_length_mm + body_mass_g

# i) Logistic regression
logMod <- glm(formula, family = "binomial", data = train)
predLog <- predict(logMod, newdata = test, type = "response")
logClass <- round(predLog)

# ii) QDA
library(MASS)
qdaMod <- qda(formula, data = train)
postQDA <- predict(qdaMod, newdata = test)$posterior
predQDA <- predict(qdaMod, newdata = test)$class

# iii) KNN
library(class)
trainKNN <- subset(train, select = -adelie)
testKNN <- subset(test, select = -adelie)
knnMod <- knn(train = trainKNN, test = testKNN, cl = train$adelie, k = 25, prob = TRUE)

# iv) Sensitivity and specificity
```

```r
# Function that calculates sensitivity and specificity from a 2x2 confusion
# matrix
sens_spec <- function(confusion_matrix, method) {
    sens <- confusion_matrix[2, 2]/(sum(confusion_matrix[2, ]))
    spec <- confusion_matrix[1, 1]/(sum(confusion_matrix[1, ]))
    cat(method, "\n")
    return(c(sensitivity = sens, specificity = spec))
}

(conf_log <- table(true = test$adelie, predicted.log = logClass))
```

```
##     predicted.log
## true  0  1
##    0 52  8
##    1  1 42
```

```r
(conf_QDA <- table(true = test$adelie, predicted.qda = predQDA))
```

```
##     predicted.qda
## true  0  1
##    0 46 14
##    1  1 42
```

```r
(conf_KNN <- table(true = test$adelie, predicted.knn = knnMod))
```

```
##     predicted.knn
## true  0  1
##    0 35 25
##    1  2 41
```

```r
sens_spec(conf_log, method = "Logistic regression")
```

```
## Logistic regression
```

```
## sensitivity specificity
##   0.9767442   0.8666667
```

```r
sens_spec(conf_QDA, method = "QDA")
```

```
## QDA
```

```
## sensitivity specificity
##   0.9767442   0.7666667
```

```r
sens_spec(conf_KNN, method = "KNN")
```

```
## KNN
```

```
## sensitivity specificity
##   0.9534884   0.5833333
```

b) (5P)

(i) Present a plot of the ROC curves and calculate the area under the curve (AUC) for each of the classifiers in a) (1P for each model).
(ii) Briefly discuss the ROC curves and the AUC. Which model performs best and worst (1P)?
(iii) If the task is to create an interpretable model, which model would you choose (1P)?

R-hints:

- To obtain $P(y = 1)$ from the `knn()` output you have to be aware that the respective probabilities

```
attributes(knnMod)$prob
```

are the success probability for the actual class where the categorization was made. So if you want to get a vector for $P(y = 1)$, you have to use $1 - P(y = 0)$ for the cases where the categorization was 0:

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

- You might find the functions `roc()` and `ggroc()` from the package `pROC` useful, but there are many ways to plot ROC curves.

Solution:

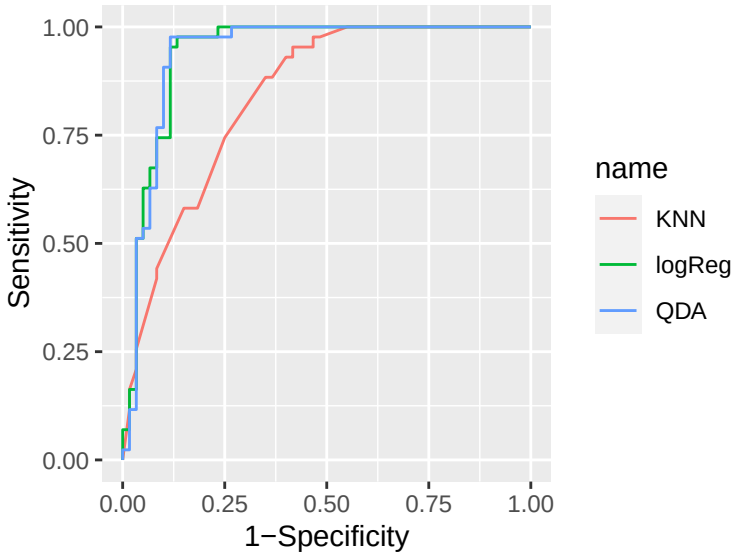
Comparison of all methods:

```
library(pROC)
library(plotROC)

logReg.ROC <- roc(response = test$adelie, predictor = predLog)
QDA.ROC <- roc(response = test$adelie, 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
# information that we need for the ROC curve:
probKNN <- ifelse(knnMod == 0, 1 - attributes(knnMod)$prob, attributes(knnMod)$prob)
KNN.ROC <- roc(response = test$adelie, predictor = probKNN)

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




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

	auc
logReg	0.9391473
QDA	0.9379845
KNN	0.8403101

(ii) Logistic regression has the highest AUC (0.5P), KNN performs worst (0.5P).
(iii) Logistic regression is also the model with the best interpretability.

c) (1P) Single choice

We are again looking at the logistic regression model that you fitted to the training data.

```
summary(logMod)$coef
```

```
##                      Estimate   Std. Error   z value     Pr(>|z|)
## (Intercept)       37.7618776 5.1761640773  7.295340 2.979055e-13
## flipper_length_mm -0.2055804 0.0324291723 -6.339367 2.307116e-10
## body_mass_g        0.0007120 0.0004619996  1.541127 1.232859e-01
```

According to this model, how would the odds that an observed animal is from the *Adelie* species change if the body mass increases by 1000 g? (the flipper length stays the same)

i) We add 0.712.
ii) We multiply it with 0.002.
iii) We multiply by 2.038.
iv) We multiply by 0.712.
v) We add 2.038.
vi) We multiply by 1000.

Solution: (iii) is correct. We know that $\beta_1 = 7.1199998 \times 10^{-4}$, thus if x_{i1} (the body mass) increases by 1000, we have to multiply the odds by $e^{\beta_1 1000} = 2.038$.

d) (2P)

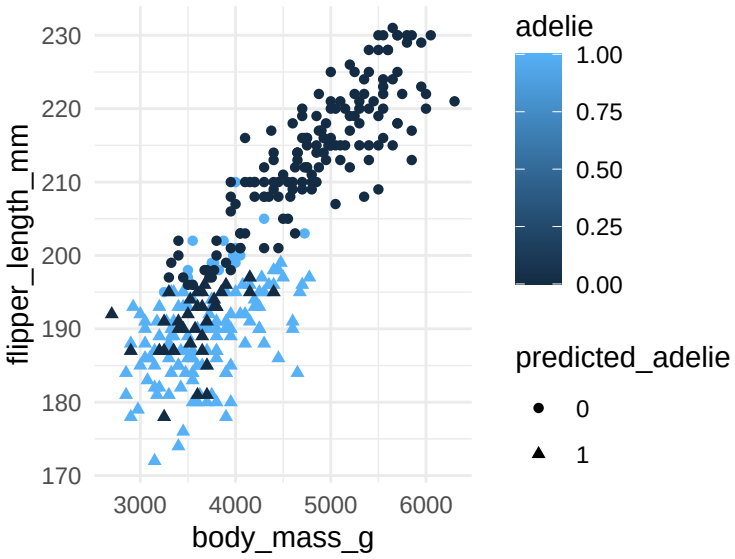
Plot the full data (including both training nd test set) with the two covariates as the x- and y-axis, and use color and some other attribute of your choice (e.g. shape or highlight) to visualize the true species (adelie/not adelie) as well as the predicted species from the best model in b) (note that the model should only be fitted with the training data as in b), but you are showing the data and predictions for both training and test data in the same plot).

Solution:

```
# IMPORTANT: We use the model from further up to predict on *the whole data set*,
# that is, including the training data.
predicted_species_prob <- predict(logMod, newdata = Penguins_reduced, type = "response")
predicted_species <- round(predicted_species_prob)

Penguins_viz <- Penguins_reduced %>% mutate(predicted_adelie = as.factor(predicted_species))

ggplot(Penguins_viz, aes(x = body_mass_g, y = flipper_length_mm)) + geom_point(aes(color = adelie,
    shape = predicted_adelie)) + theme_minimal()
```



Problem 4 (10P)

a) (2P) - Multiple choice

Which statements about validation set approach, k-fold cross-validation (CV) and leave-one-out cross validation (LOOCV) are true and which are false? Say for *each* of them if it is true or false.

(i) The validation set-approach is computationally cheaper than 10-fold CV.
(ii) 5-fold CV will generally lead to less bias, but more variance than LOOCV in the estimated prediction error.
(iii) The validation set-approach is the same as 2-fold CV.
(iv) LOOCV is always the cheapest way to do cross-validation.

Solution:

TRUE, FALSE, FALSE, FALSE

(i) is correct. In (ii) it would be the other way round. (iii) is wrong, because in 2-fold CV we would fit the model twice (once with each half of the data), while the validation set approach uses only one half of the data to fit the model. (iv) is wrong, because LOOCV is actually very expensive - expect for linear regression, where a formula exists.

b) (2P)

We are now looking at a bootstrap example. Assume you want to fit a model that predicts the probability for coronary heart disease (`chd`) from systolic blood pressure (`sbp`), sex (0=female, 1=male) and smoking status (0=no, 1=yes). Load the data in R as follows

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

and perform a logistic regression with `chd` as outcome and `sbp`, `sex` and `smoking` as covariates. What is the probability of chd for a non-smoking male with a sbp=150 in the given dataset?

Solution: 1P for doing the regression, 1P for calculating the probability.

```r
id <- "1chRpybM5cJn4Eow3-_xwDKPKyddL9M2N"  # google file ID
d.chd <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download", id))
r.glm <- glm(chd ~ sbp + sex + smoking, d.chd, family = "binomial")
summary(r.glm)
```

```
##
## Call:
## glm(formula = chd ~ sbp + sex + smoking, family = "binomial",
##     data = d.chd)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0184  -0.5950  -0.3790  -0.2954   2.5570
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.65884    2.36740  -2.813  0.00491 **
## sbp          0.03877    0.01794   2.162  0.03066 *
## sex         -1.34351    0.32148  -4.179 2.93e-05 ***
## smoking      0.41031    0.31014   1.323  0.18584
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 370.89  on 499  degrees of freedom
## Residual deviance: 342.91  on 496  degrees of freedom
## AIC: 350.91
##
## Number of Fisher Scoring iterations: 5
```

Deriving the probability:

To get the probability you can now either plug in the estimates and calculate $p = exp(\eta)/(1 + exp(\eta))$ with $\eta = \beta_0 + \beta_1 \cdot 150 + \beta_2 \cdot 1 + \beta_3 \cdot 0$, or directly use the predict function in R:

```r
newdata <- data.frame(sbp = 150, sex = 1, smoking = 0)
(pred.p <- predict(r.glm, newdata, type = "response"))
```

```
##       1
## 0.10096
```

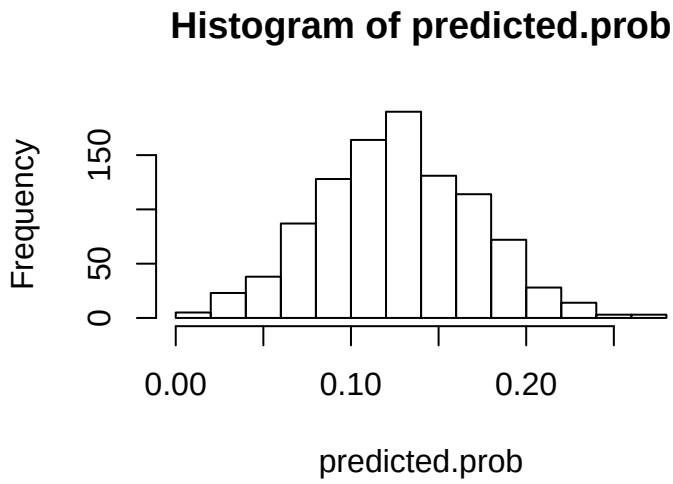
c) (4P)

We now use the bootstrap to estimate the uncertainty of the probability derived in b). Use $B = 1000$ bootstrap samples and proceed as follows:

- In each iteration, derive and store the estimated probability for `chd`, given `sbp`=150, `sex`=male and `smoking`=0 (1P for implementing the bootstrap).
- From the set of estimated probabilities, derive the standard error (1P).
- Derive the 95% quantile interval for the bootstrap samples (that is, the interval with limits at 2.5% and 97.5%) (1P).
- Interpret what you see. What is the expected probability and what are plausible values? (1P)

Solution:

```
B <- 1000
predicted.prob <- rep(NA, B)
for (ii in 1:B) {
    d.boot <- d.chd[sample(1:nrow(d.chd), size = nrow(d.chd), replace = TRUE), ]
    r.glm.boot <- glm(chd ~ sbp + sex + smoking, data = d.boot)
    newdata <- data.frame(sbp = 150, sex = 1, smoking = 0)
    predicted.prob[ii] <- predict(r.glm.boot, newdata, type = "response")
}
hist(predicted.prob)
```



The standard error is just the standard deviation of the sample. The 95% CI can be obtained by using the interval ranging from the 2.5% to the 97.5% quantile of the samples.

```
sd(predicted.prob)
```

```
## [1] 0.04493216
```

```
quantile(predicted.prob, probs = c(0.025, 0.975))
```

```
##       2.5%      97.5%
## 0.03813944 0.21554645
```

Interpretation: The predicted probability for chd for a non-smoking male with sbp=150 is 0.101, the respective standard error is 0.045 and the 95% CI, which contains the range of plausible values, is ranging from 0.038 to 0.216.

d) Multiple choice - 2P

We continue with the same dataset to study some properties of the bootstrap method. Below we estimated the standard errors of the regression coefficients in the logistic regression model with `sex`, `sbp` and `smoking` as predictors using 1000 bootstrap iterations (column `std.error`). These standard errors can be compared to those that we obtain by fitting a single logistic regression model using the `glm()` function (in Problem 4b). Look at the R output below and compare the standard errors that we obtain from the bootstrap with those we get from the `glm()` function (note that the `t1*` to `t4*` variables are sorted in the same way as for the `glm()` output).

```
set.seed(4268)
library(boot)
boot.fn <- function(data, index) {
    return(coefficients(glm(chd ~ sbp + sex + smoking, family = "binomial", data = data,
```

```
        subset = index)))
}
boot(d.chd, boot.fn, 1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = d.chd, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##        original        bias    std. error
## t1* -6.65883685  0.0142486669  2.54155686
## t2*  0.03877165 -0.0001413491  0.01921332
## t3* -1.34351384 -0.0462122125  0.33868464
## t4*  0.41031080 -0.0220322661  0.33473663
```

Which of the following statements are true? Say for *each* of them if it is true or false.

(i) The bootstrap relies on random sampling the same data without replacement.
(ii) The estimated standard errors from the `glm()` function are smaller than those estimated from the bootstrap, which indicates a problem with the bootstrap.
(iii) In general, differences between the estimated standard errors from the bootstrap and those from `glm()` may indicate a problem with the assumptions taken in logistic regression.
(iv) The p-values from the `glm()` output are probably slightly too small.

Solution:

FALSE - FALSE - TRUE - TRUE

(i) No, it is with replacement.
(ii) No, the bootstrap is "always right"
(iii) Yes, because (ii) is false.
(iv) Yes, because the SEs are a bit too small.