Compulsory exercise 1: Group 25

TMA4268 Statistical Learning V2022

Plato Karageorgis, Jonas Gustav Dønheim Nordstrøm, Fanny Øverbø Næss and Ole Kristian Skogly

20 februar, 2022

Problem 1

a)

$$\begin{aligned}\text{MSE} &= E[(y-\hat{f}(x))^2] \\ &= E[(\hat{f}(x)^2 - 2E[\hat{f}(x)y]] + E[y^2]\end{aligned}$$

We substitute y with $f(x)+\epsilon$

$$= E[(\hat{f}(x)^2 - 2E[\hat{f}(x)(f(x)+\epsilon)]] + E[(f(x)+\epsilon)^2]$$

Then by expanding and using that ϵ is independent from $f(x)$ and $\hat{f}(x)$, we find

$$= E[\hat{f}(x)^2] - 2\left(E[\hat{f}(x)]f(x)) + E[\epsilon]E[\hat{f}(x)]\right) + f(x)^2 + 2E[\epsilon]f(x) + E[\epsilon^2]$$

We assume that $\epsilon \sim N(0,\sigma_\epsilon^2)$, then

$$E[\epsilon]=0, \qquad \text{Var}[\epsilon] = E[\epsilon^2] = \sigma_\epsilon^2$$

And we have

$$\text{Bias}[\hat{f}(x)] = E[\hat{f}(x)] - f(x), \qquad \text{Var}(\hat{f}(x)) = E[(\hat{f}(x)] - E[\hat{f}(x)])^2]$$

Using this and by adding and subtracting $E[\hat{f}(x)]^2$ from our expression for the MSE we get

$$\begin{aligned}&= \left(E[\hat{f}(x)]^2 - 2E[\hat{f}(x)]f(x) + f(x)^2\right) + E[\hat{f}(x)^2] - E[\hat{f}(x)]^2 + Var[\epsilon] \\ &= \left(E[\hat{f}(x)] - f(x)\right)^2 + \text{Var}[\hat{f}(x)] + \sigma_\epsilon^2\end{aligned}$$

$$= \text{Bias}[\hat{f}(x)]^2 + \text{Var}[\hat{f}(x)] + \sigma_\epsilon^2$$

b)

As denoted in a), the bias is defined as the square error between the expected value of $\hat{f}$ and the true function f. The bias decreases when the complexity of $\hat{f}$ increases. The more complex $\hat{f}$ is, the closer it may approximate each data point y_i. However, the reduction of bias as denoted in **a)**, the bias is defined as the difference between the expected value of $\hat{f}$ and the true function f. The bias decreases when the complexity of $\hat{f}$ increases. The more complex $\hat{f}$ is, the closer it may approximate each data point y_i.

The variance is a measure of the spread of the outputs of $\hat{f}$. The variance increases with the complexity of $\hat{f}$, and so, there is a trade-off between variance and bias.

The irreducible error is the variance of the *true* error, ϵ. The error is independent of the model, and is therefore irreducible.

c)

- i) True
- ii) False
- iii) True
- iv) False

d)

- i) True
- ii) False
- iii) True
- iv) False

e)

- iii) 0.76

Problem 2

a)

- **Removes sex as a variable:** We were given a model developed by experts, and should not remove any elements from it. Additionally, he has misunderstood the meaning of p-values. Because sex has the smallest p-value, it is the most significant, and would therefore lower the quality of the prediction significantly if removed. If one were to remove a variable to avoid overfitting, one should remove the variable that is the least significant.
- **False statement about coefficients:** Basil states that "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." We do not know anything of the relationship between the coefficients speciesChinstrap and speciesGentoo because of the way that the dummy variables has been coded.

```
attach(penguins)
contrasts(species)
```

```
##           Chinstrap Gentoo
## Adelie            0      0
## Chinstrap         1      0
## Gentoo            0      1
```

From this we see that a binary relationship has been implemented for the three classes, where we have two pairwise comparisons between Adelie and the two other species. Because of this, we never compare Chinstrap and Gentoo with each other, and can therefore not determine whether there is a significance between all species.

- **Misunderstands coefficients:** Basil states that we can tell that the Chinstrap has the largest body mass based on the coefficient $\hat{\beta}_{chinstrap}$. However, this coefficient is only related to Adelie, and not Gentoo. Since the dummy variable has more than two levels, it can not represent all possible variables.

b)

```
library(palmerpenguins)
library(GGally)
summary(penguins)
```

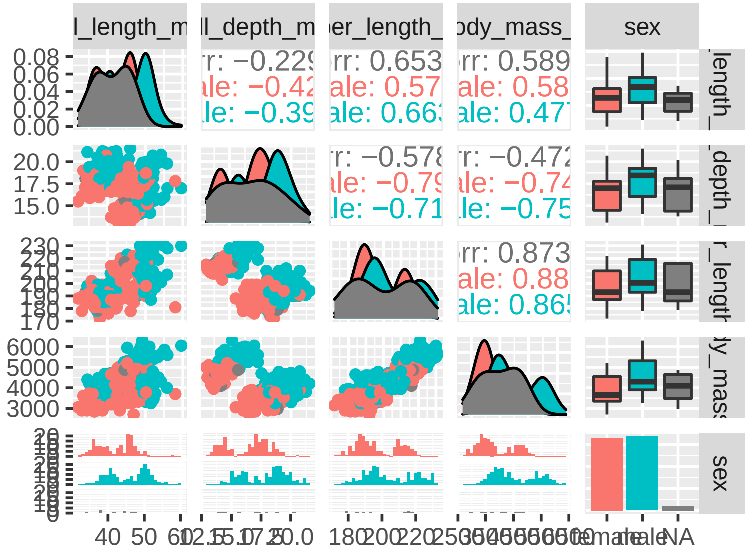
```
##       species          island    bill_length_mm  bill_depth_mm
##  Adelie   :152   Biscoe   :168   Min.   :32.10   Min.   :13.10
##  Chinstrap: 68   Dream    :124   1st Qu.:39.23   1st Qu.:15.60
##  Gentoo   :124   Torgersen: 52   Median :44.45   Median :17.30
##                                  Mean   :43.92   Mean   :17.15
##                                  3rd Qu.:48.50   3rd Qu.:18.70
##                                  Max.   :59.60   Max.   :21.50
##                                  NA's   :2       NA's   :2
##  flipper_length_mm  body_mass_g       sex           year
##  Min.   :172.0     Min.   :2700   female:165   Min.   :2007
##  1st Qu.:190.0     1st Qu.:3550   male  :168   1st Qu.:2007
##  Median :197.0     Median :4050   NA's  : 11   Median :2008
##  Mean   :200.9     Mean   :4202                Mean   :2008
##  3rd Qu.:213.0     3rd Qu.:4750                3rd Qu.:2009
##  Max.   :231.0     Max.   :6300                Max.   :2009
##  NA's   :2         NA's   :2
```

```
names(penguins)
```

```
## [1] "species"           "island"            "bill_length_mm"
## [4] "bill_depth_mm"     "flipper_length_mm" "body_mass_g"
## [7] "sex"               "year"
```

```
#Select a subset of variables that are continous
penguins.plot<-subset(penguins, select = -c(island,species,year))
#A subset of sex and body_mass_g
penguins.sex<-subset(penguins, select = c(sex,body_mass_g))

#Plot of continuous variables with sex coloring, male=blue, female=red
ggpairs(penguins.plot, aes(color = sex))
```



c)

```
penguin.model <- lm(body_mass_g ~ flipper_length_mm + sex + bill_depth_mm * species, data = penguins)
summary(penguin.model)
```

```
##
## Call:
## lm(formula = body_mass_g ~ flipper_length_mm + sex + bill_depth_mm *
##     species, data = 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 ***
## bill_depth_mm                   82.98      22.32   3.717 0.000237 ***
## speciesChinstrap              1460.15     680.39   2.146 0.032610 *
## speciesGentoo                  644.88     542.57   1.189 0.235481
## bill_depth_mm:speciesChinstrap -83.53      37.01  -2.257 0.024666 *
## bill_depth_mm:speciesGentoo     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
##   (11 observations deleted due to missingness)
## Multiple R-squared:  0.8758, Adjusted R-squared:  0.8732
## F-statistic: 327.5 on 7 and 325 DF,  p-value: < 2.2e-16
```

According to the exercise description, Basil was given an optimal model developed by experts, hence it is unnecessary to investigate the model for optimization. Moreover, in comparison with Basil's report, we will not exclude the sex variable because we observe from the p-value that it is significant. The only predictor that has a high p-value is the species' category speciesGentoo and subsequently bill_depth_mm interaction with it, but we should not remove the species variable since it is a crucial part of our model. The model can be described by these functions:

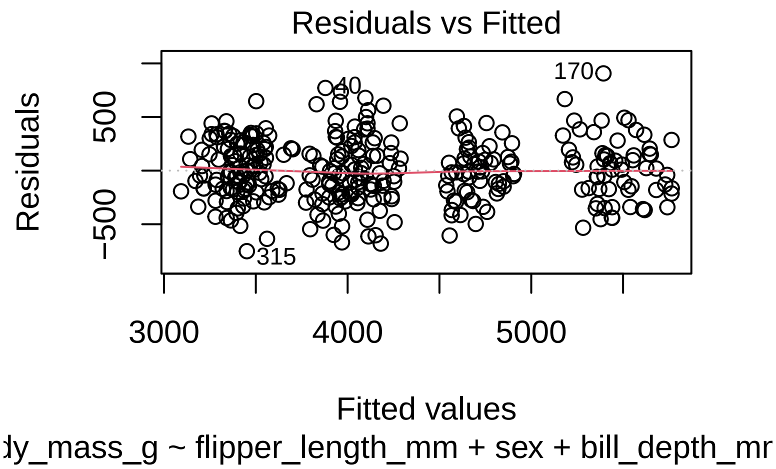
$$\hat{y}_{adelie} = \hat{\beta}_0 + \hat{\beta}_{flipperlength} x_{flipperlength} + \hat{\beta}_{billdepth} x_{billdepth} + \hat{\beta}_{sex} x_{sex}$$

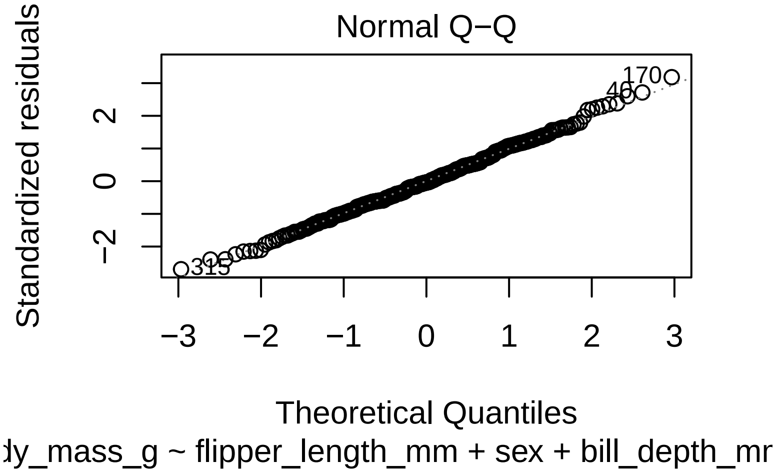
$$\hat{y}_{chinstrap} = \hat{\beta}_0 + \hat{\beta}_{flipperlength} x_{flipperlength} + (\hat{\beta}_{billdepth} + \hat{\beta}_{billdepth:chinstrap}) x_{billdepth} + \hat{\beta}_{chinstrap} + \hat{\beta}_{sex} x_{sex}$$

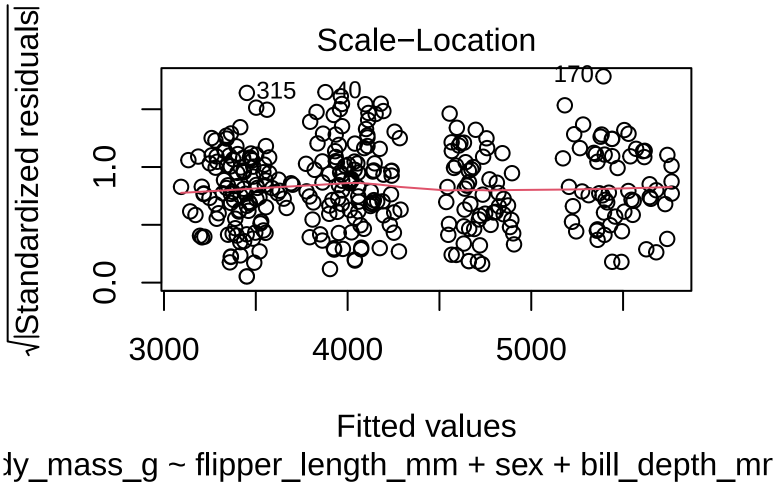
$$\hat{y}_{gentoo} = \hat{\beta}_0 + \hat{\beta}_{flipperlength} x_{flipperlength} + (\hat{\beta}_{billdepth} + \hat{\beta}_{billdepth:gentoo}) x_{billdepth} + \hat{\beta}_{gentoo} + \hat{\beta}_{sex} x_{sex}$$

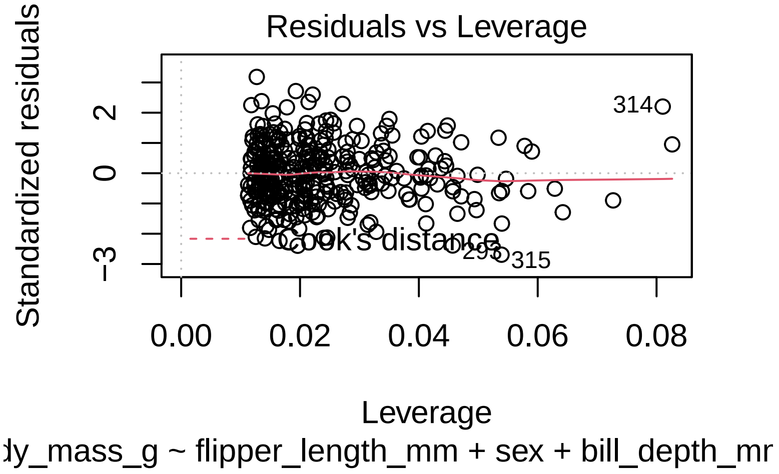
We note that the coefficient x_{sex} is binary, 1 if male, 0 if female. After the summary, due to the very large difference in the Chinstrap species compared to others and also the difference of the male penguins, we have the intuition that these have the bigger body mass but we can't be certain because we need more extensive analysis and different tests like anova.

```
plot(penguin.model)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
−3
−2
−1
0
1
2
3
−2
0
2
170
40
315
dy_mass_g ~ flipper_length_mm + sex + bill_depth_mr


Scale−Location
√|Standardized residuals|
Fitted values
3000
4000
5000
0.0
1.0
315
40
170
dy_mass_g ~ flipper_length_mm + sex + bill_depth_mr



From the residual plot we see a random spread, indicating that the linear model fits the data well, and therefor that we don't need a more complex model to fit the data. The Q-Q plot strongly suggests that our assumption about the data being Normal distributed is correct.

Problem 3

a)

Firstly, we need to run the code that was prepared for this task

```
# 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)) %>%di
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, ]
```

Then, we use each of the classification methods. Firstly, we use logistic regression from the glm package

```
#Logistic regression
fit.log<-glm(adelie~.,family = binomial, data = train)
summary(fit.log)
```

```
##
## Call:
```

```
## glm(formula = adelie ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6506  -0.4133  -0.1161   0.6550   2.2962
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       37.761878   5.176164   7.295 2.98e-13 ***
## body_mass_g        0.000712   0.000462   1.541    0.123
## flipper_length_mm -0.205580   0.032429  -6.339 2.31e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 329.11  on 238  degrees of freedom
## Residual deviance: 184.21  on 236  degrees of freedom
## AIC: 190.21
##
## Number of Fisher Scoring iterations: 6
```

```
log.probs <- predict(fit.log,test, type ="response")
log.preds <- ifelse(log.probs>0.5, 1,0)
#Confusion matrix
log.matrix<-table(log.preds,test$adelie)
cat("Confusion matrix:")
```

```
## Confusion matrix:
```

```
log.matrix
```

```
##
## log.preds  0  1
##         0 52  1
##         1  8 42
```

Then, we fit the qda model

```
fit.qda <- qda(adelie~.,data = train)
qda.preds <- predict(fit.qda,newdata = test)$class
qda.probs <- predict(fit.qda,newdata = test)$posterior
qda.matrix<-table(qda.preds,test$adelie)
cat("Confusion matrix:")
```

```
## Confusion matrix:
```

```
qda.matrix
```

```
##
## qda.preds  0  1
##         0 46  1
##         1 14 42
```

Lastly, the knn model with k=25

```
set.seed(7)
fit.knn<- knn(train = train, test = test, cl = train$adelie, k = 25, prob = T )
knn.probs <- ifelse(fit.knn == 0, 1 - attributes(fit.knn)$prob, attributes(fit.knn)$prob)
knn.matrix<-table(fit.knn,test$adelie)
cat("Confusion matrix:")
```

```
## Confusion matrix:
```

```
knn.matrix
```

```
##
## fit.knn  0  1
##       0 35  2
##       1 25 41
```

Lastly we calculate the specificity and sensitivity for each of the methods. To do this more easily, we define a function that calculates this, using the confusion matrix.

```
sensitivity<-function(matrix){
  return(matrix[2,2]/(matrix[1,2]+matrix[2,2]))
}
specificity<-function(matrix){
  return(matrix[1,1]/(matrix[1,1]+matrix[2,1]))
}
```

So for logistic regression we have

```
cat("the sensitivity of logistic regression is:",sensitivity(log.matrix),"\n")
```

```
## the sensitivity of logistic regression is: 0.9767442
```

```
cat("the specificity of  logistic regression is:",specificity(log.matrix))
```

```
## the specificity of  logistic regression is: 0.8666667
```

Similarly, for qda

```
cat("the sensitivity of qda is:",sensitivity(qda.matrix),"\n")
```

```
## the sensitivity of qda is: 0.9767442
```

```
cat("the specificity of qda is:",specificity(qda.matrix))
```

```
## the specificity of qda is: 0.7666667
```

Lastly, for the knn

```
cat("the sensitivity of knn is:",sensitivity(knn.matrix),"\n")
```

```
## the sensitivity of knn is: 0.9534884
```

```
cat("the specificity of knn is:",specificity(knn.matrix))
```

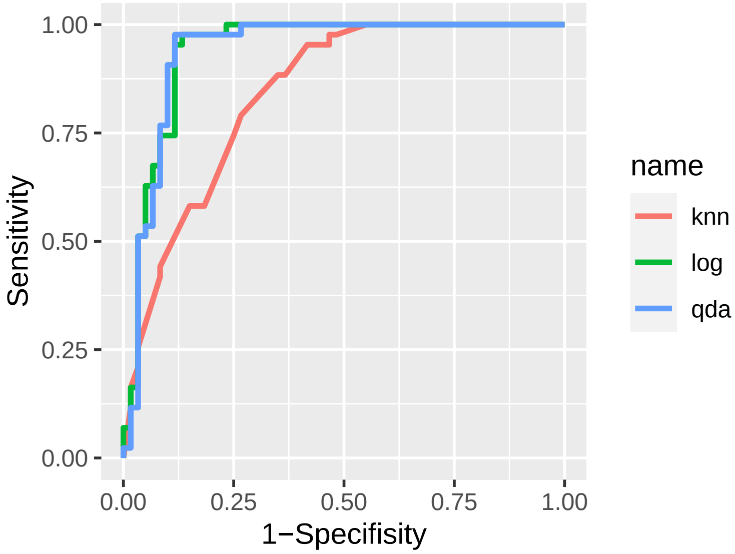
```
## the specificity of knn is: 0.5833333
```

b)

Firstly we find the roc curve for each of the models

```
logroc <- roc(response = test$adelie, predictor = log.probs, direction = "<")
qdaroc <- roc(response = test$adelie, predictor = qda.probs[,2], direction = "<")
knnroc <- roc(response = test$adelie, predictor = knn.probs, direction = "<")

dat <- data.frame(adelie = test$adelie, log = log.probs, qda = qda.probs[,2], knn = knn.probs)
dat.long <- melt_roc(dat, "adelie", c("log","qda","knn"))
ggplot(dat.long, aes(d=D, m=M, color = name))+geom_roc(n.cuts =F) + xlab("1-Specifisity")+ ylab("Sensiti
```



Now, we can find the area under the roc curve (auc)

```
cat("The auc of the logistic regression is:",auc(logroc),"\n")
```

```
## The auc of the logistic regression is: 0.9391473
```

```
cat("The auc of the qda is:",auc(qdaroc),"\n")
```

```
## The auc of the qda is: 0.9379845
```

```
cat("The auc of the knn is:",auc(knnroc))
```

```
## The auc of the knn is: 0.8416667
```

From the curves and the auc values we may conclude that logistic regression and qda perform very similar; where logistic regression has a slight edge. Knn performs comperatively poorly to the other methods, though much better than chance.

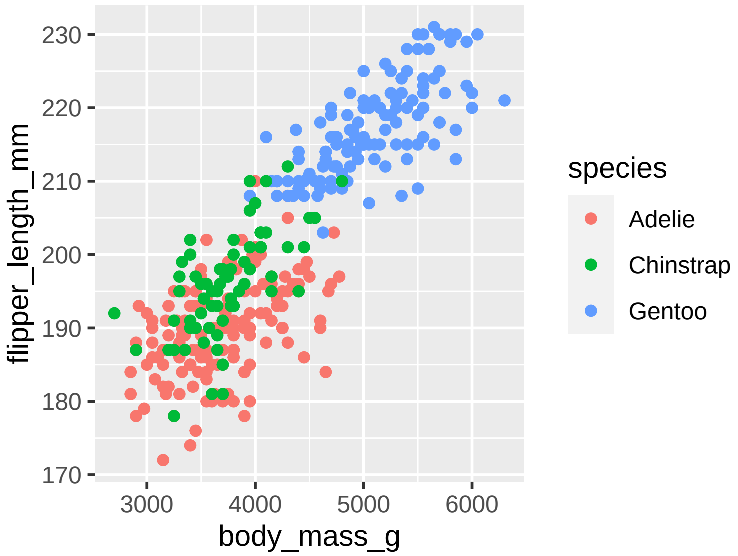
Logistic regression has the advantage of being able to interpret, since we know to what scale each covariate interact with the response by looking at their coefficients. This, in addition to being the best predictive model, makes it the best model to use.

c)

The correct option is: (iii) We multiply by 2.038.

d)

```
penguins.sub<- subset(penguins,select = c(body_mass_g,flipper_length_mm,species))
ggpairs(penguins.sub,aes(color = species))[2,1]
```



Problem 4

a)

- i) True
- ii) False
- iii) False
- iv) False

b)

```r
id <- "1chRpybM5cJn4Eow3-_xwDKPKyddL9M2N" # google file ID
d.chd <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download", id))
#Fitting a regression and using it to predict the probability
fit.chd<-glm(chd~.,family = binomial,data=d.chd)
summary(fit.chd)
```

```
##
## Call:
## glm(formula = chd ~ ., 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 **
## sex         -1.34351    0.32148  -4.179 2.93e-05 ***
## sbp          0.03877    0.01794   2.162  0.03066 *
## 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
```

```r
probchd<-predict(fit.chd,data.frame(sex = 1, sbp = 150, smoking=0),type= "response")
cat("The probability that a non-smoking male with a sbp=150 in the given dataset gets chd is",probchd)
```

```
## The probability that a non-smoking male with a sbp=150 in the given dataset gets chd is 0.10096
```

c)

We chose to implement the bootstrap ourself instead of using the boot function.

```r
#Defining the function bootstrapping samples from
#Return the predicted probability of cdf given the parameters
boot.fn<-function(data,index){
  fit.chd<- glm(chd~.,family = binomial, data = d.chd, subset = index)
  return(predict(fit.chd,data.frame(sex = 1, sbp = 150, smoking=0),type= "response"))
}
#Own implementation of boot
prob.vec<-c()
for(i in 1:1000){
  index <- sample(dim(d.chd)[1],dim(d.chd)[1],replace = T)
  prob.vec[i]<-boot.fn(d.chd,index)
}
cat("Predicted probability from the bootstrap is:",mean(prob.vec),"\n")
```

```
## Predicted probability from the bootstrap is: 0.1060886
```

```
cat("Predicted standard error form the bootstrap is:",sd(prob.vec),"\n")
```

```
## Predicted standard error form the bootstrap is: 0.04404331
```

```
cat("The 95% quantiles is:\n")
```

```
## The 95% quantiles is:
```

```
quantile(prob.vec,c(0.0275,0.975))
```

```
##      2.75%      97.5%
## 0.03796642 0.20471663
```

d)

- i) False
- ii) False
- iii) True
- iv) True