Compulsory exercise 1: Group 18

TMA4268 Statistical Learning V2022

Erlend Nonås Lokna, Thomas Fardal Rødland

03 September, 2023

Problem 1

a)

The expected test MSE, $E[(y_0 - \hat{f}(x_0))^2]$, can be written as $E[f(x_0) + \epsilon_0 - f(\hat{x_o})]$. We then get the following calculation

$$E[(f(x_0) - f(\hat{x_0}))^2] + E[\epsilon^2] + 2E[(f(x_0) - f(\hat{x_o})] * E[\epsilon]$$

Now since the expectation of ϵ is zero we get

$$= E[(f(x_0) - f(\hat{x_0}))^2] + E[\epsilon^2]$$

$$= E[(f(x_0) - f(\hat{x_0}))^2] + Var(\epsilon)$$

$$= E[(f(x_0) - E[\hat{(}f(x_0))] - (f(\hat{x_0}) - E[f(\hat{x_0})]))^2] + Var(\epsilon)$$

$$= E[(f(\hat{x_0}) - f(x_0))]^2 + E[(f(\hat{x_0}) - E[f(\hat{x_0})])^2] + Var(\epsilon)$$

$$= bias(\hat{f}) + Var(\hat{f}) + (\sigma_\epsilon))^2$$

b)

The three terms bias, variance and irreducible error can be interpreted as follows.

Irreducible error: An error that is inherent within the problem itself and therefore something that we can not minimize. It is an error that will always be there.

Bias: The error we get from our modeling assumptions. If we pick a model with incorrect assumptions towards our data set we can get a high bias and an underfitting of our problem.

Variance: The variance is the error that occurs when our model is to sensitive to small fluctuations in our training data. This can cause an overfitting, and our model will may not work as well as intended for other data sets. To reduce the error in our modelling we try to minimize bias and variance.

c)

I) True
II) False
III) True
IV) False

d)

I) True
II) False
III) True
IV) False

e)

The formula for correlation is given as

$$cor(x_1, x_2) = \frac{cov(x_1, x_2)}{\sqrt{\sigma_1}\sqrt{\sigma_2}}$$

which in our case is

$$\frac{33}{\sqrt{50}\sqrt{38}} = 0.76$$

Problem 2

```
# 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)

final.model <- lm(body_mass_g ~ flipper_length_mm + bill_depth_mm * species, data = penguins)
summary(final.model)$r.squared
```

```
## [1] 0.8364086
```

a)

- The formulas for the multi-linear regression lacks the error/residual terms. The error is a important factor to the analysis.

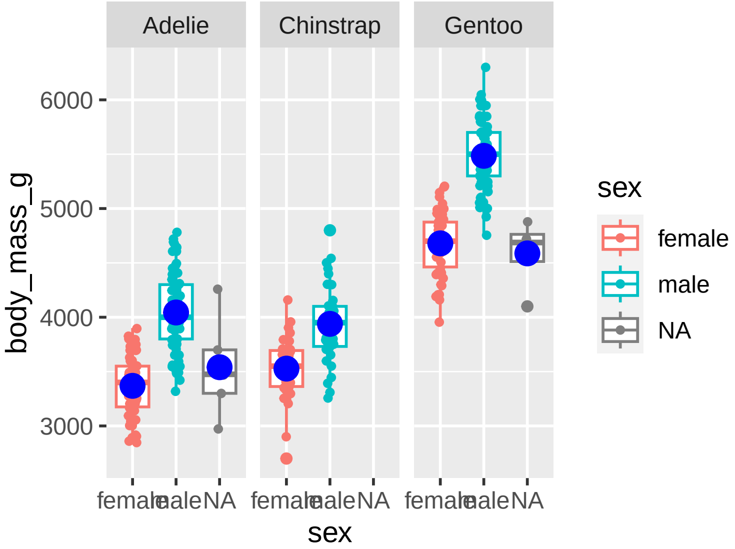
- It is wrong to assume that the chinstrap penguins have the larges body mass based on the coefficient alone.

- Basil excludes the sex covariate in the final model for no good reason or argumentation. The male-sex covariate has the lowest p-value making it a big mistake to exclude it in the final model.

b)

Basil's final model excludes the penguins gender without much argumentation. As you can see in the following plot it is clear that the gender variable should not be neglected as the male penguins clearly has more body weight compared to the female penguins. This is a trend for all the penguin species.

```
ggplot(penguins, aes(x = sex, y = body_mass_g, color = sex)) +
    geom_boxplot()+
    geom_point(size = 1.0, position = position_jitter(width = 0.1)) +
    stat_summary(fun.y = mean, geom = "point", shape = 20, size = 6, color = "blue")+
    facet_grid(.~species)
```

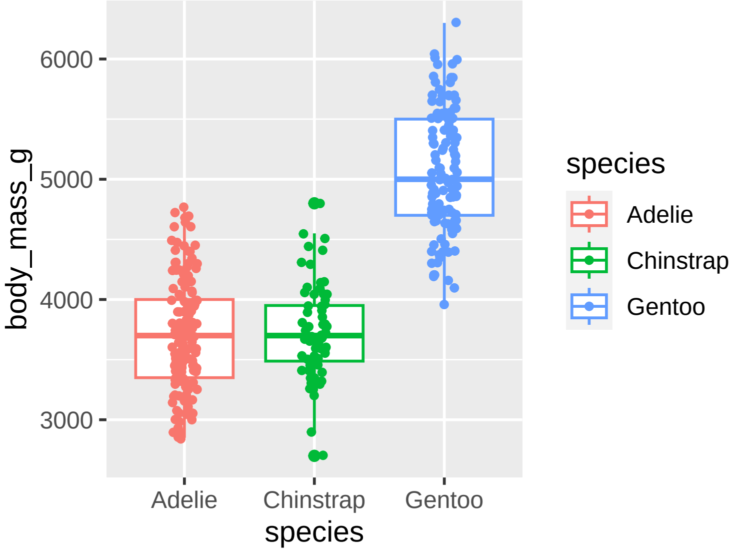


The R-squared value in Basil's final model has a value of 0.83. Now, looking at the final model which includes the sex (see 2c) we find a resulting value of 0.88 which proves that including the gender results in a better model fit.

Basil also excludes the island and the year without argumentation, even though it was provided by expert knowledge. This is a mistake, since you can see that the year-covariate has a p-value close to but yet below 0.05. Also, to exclude covariates without argumentation or validation is not good.

It is wrong to say that the chinstrap penguins has the highest body mass based on the coefficient $\beta_{chinstrap}$ alone, as you can see in the following figure:

```r
ggplot(penguins, aes(x = species, y = body_mass_g, color = species)) +
  geom_boxplot()+
   geom_point(size = 1.0, position = position_jitter(width = 0.1))
```



The Gentoo penguins has the largest average body mass among the penguin species.

c)

We start by making a linear model with all of the covariates given by the experts. Then we will examine the fit of the model and which covariates to potentially exclude.

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

```
##
## Call:
## lm(formula = body_mass_g ~ flipper_length_mm + island + year +
##     sex + bill_depth_mm * species, data = penguins)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -760.0 -200.3  -12.7  174.6  853.5
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    89562.562  42170.810   2.124  0.03445 *
## flipper_length_mm                 20.019      3.115   6.427 4.69e-10 ***
## islandDream                      -30.427     58.130  -0.523  0.60104
## islandTorgersen                  -62.677     60.710  -1.032  0.30265
## year                             -45.423     21.059  -2.157  0.03175 *
## sexmale                          422.540     44.752   9.442  < 2e-16 ***
## bill_depth_mm                     74.630     22.707   3.287  0.00113 **
## speciesChinstrap                1438.772    678.374   2.121  0.03469 *
## speciesGentoo                    384.748    554.498   0.694  0.48827
## bill_depth_mm:speciesChinstrap   -83.433     36.914  -2.260  0.02448 *
## bill_depth_mm:speciesGentoo       44.811     34.709   1.291  0.19760
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 285.7 on 322 degrees of freedom
##   (11 observations deleted due to missingness)
## Multiple R-squared:  0.8779, Adjusted R-squared:  0.8741
## F-statistic: 231.5 on 10 and 322 DF,  p-value: < 2.2e-16
```

By looking at the respective p-values in the summary, we see that the islandDream covariate has a high p-value (>0.05) indicating that it can be removed from the final model without it drastically effecting the model fit. The islandTorgersen has a lower p-value of 0.03. However it can still be considered for removal. Since both of the islands offer a high p-value, we will exclude the island covariate in the final model. It is worth mentioning that the most influential (dominant) covariates in the model is the flipper_length_mm, male-sex and bill_depth_mm.

By examining the data collected through the years, we can see that the average values is similar for each year, however the variance and number of observations is changing. This is especially clear for the Chinstrap penguins. Since the p-value of the year covariate is below 0.05 it should stay in the model to improve the overall r-squared value.

We will now continue to make the final model after examining the summary. To conclude upon the examination of the individual covariates, we would like to include everything except the island in the final model.

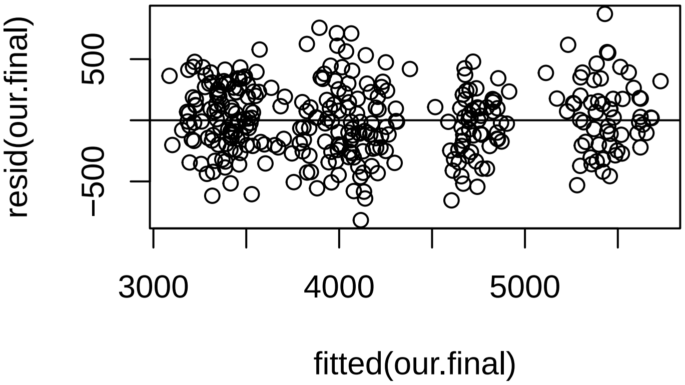
```
our.final = lm(body_mass_g ~ flipper_length_mm + year + sex + bill_depth_mm + species, data = penguins)
summary(our.model)
```

```
##
```

```
## Call:
## lm(formula = body_mass_g ~ flipper_length_mm + island + year +
##     sex + bill_depth_mm * species, data = penguins)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -760.0 -200.3  -12.7  174.6  853.5
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     89562.562  42170.810   2.124  0.03445 *
## flipper_length_mm                  20.019      3.115   6.427 4.69e-10 ***
## islandDream                       -30.427     58.130  -0.523  0.60104
## islandTorgersen                   -62.677     60.710  -1.032  0.30265
## year                              -45.423     21.059  -2.157  0.03175 *
## sexmale                           422.540     44.752   9.442  < 2e-16 ***
## bill_depth_mm                      74.630     22.707   3.287  0.00113 **
## speciesChinstrap                 1438.772    678.374   2.121  0.03469 *
## speciesGentoo                     384.748    554.498   0.694  0.48827
## bill_depth_mm:speciesChinstrap    -83.433     36.914  -2.260  0.02448 *
## bill_depth_mm:speciesGentoo        44.811     34.709   1.291  0.19760
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 285.7 on 322 degrees of freedom
##   (11 observations deleted due to missingness)
## Multiple R-squared:  0.8779, Adjusted R-squared:  0.8741
## F-statistic: 231.5 on 10 and 322 DF,  p-value: < 2.2e-16
```

As seen in the summary, our model has a better R-squared and adjusted R-squared value. Indicating that the model has a better fit, compared to Basil the cat's model. To visualize the fit graphically, we can examine a residual plot.

```r
library(cowplot)



res_plot <- plot(fitted(our.final),resid(our.final)) + abline(0,0)
```



```r
sd_res <- sqrt(deviance(our.final)/df.residual(our.final))
sd_res
```

```
## [1] 288.7706
```

The residual plot shows that the model predicts with an error in the range of $\pm500g$. Which is not too bad. By looking at the density of the residuals it clearly resembles the bell shaped normal distribution. Indicating that the error terms in the model is normally distributed (which should be the case) with mean 0 and standard deviation 288.8g. This can also be validated by a QQ-plot, where you can see that the residuals is clearly normal distributed by the fact that they are on the line.

Problem 3

```r
#Problem 3)
#a i
# 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) #Setting a seed so we can replicate our results.

# 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 logistic regression
log.fit <- glm(adelie~., data = train, family= binomial(logit))

#Using our model on the test data
log.fit.p = predict(log.fit, newdata = test, type = 'response')

testclass= ifelse(log.fit.p > 0.5,1,0) # classification with .5 cutoff
confusionMatrix(as.factor(testclass), reference = as.factor(test$adelie),positive='1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 52  1
##          1  8 42
##
##                Accuracy : 0.9126
##                  95% CI : (0.8406, 0.9593)
##     No Information Rate : 0.5825
##     P-Value [Acc > NIR] : 9.666e-14
```

```
##
##                   Kappa : 0.8244
##
##  Mcnemar's Test P-Value : 0.0455
##
##             Sensitivity : 0.9767
##             Specificity : 0.8667
##          Pos Pred Value : 0.8400
##          Neg Pred Value : 0.9811
##              Prevalence : 0.4175
##          Detection Rate : 0.4078
##    Detection Prevalence : 0.4854
##       Balanced Accuracy : 0.9217
##
##        'Positive' Class : 1
##
```

```r
#Using confusion matrix to get the statistics that we want.
```

```r
#a
#ii)

# A QDA model
qda.fit <- qda(adelie~., data = train)
qda.fit.p = predict(qda.fit ,newdata = test)$posterior[,2]
testclass_ = ifelse(qda.fit.p > 0.5,1,0)#Classification with .5 cutoff
confusionMatrix(as.factor(testclass_),reference = as.factor(test$adelie),positive='1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 46  1
##          1 14 42
##
##                Accuracy : 0.8544
##                  95% CI : (0.7712, 0.9161)
##     No Information Rate : 0.5825
##     P-Value [Acc > NIR] : 2.407e-09
##
##                   Kappa : 0.7129
##
##  Mcnemar's Test P-Value : 0.001946
##
##             Sensitivity : 0.9767
##             Specificity : 0.7667
##          Pos Pred Value : 0.7500
##          Neg Pred Value : 0.9787
##              Prevalence : 0.4175
##          Detection Rate : 0.4078
##    Detection Prevalence : 0.5437
##       Balanced Accuracy : 0.8717
##
##        'Positive' Class : 1
##
```

```r
#a
#iii)
knn.fit= knn(train = train, test = test, cl = train$adelie, k = 25, prob = T)
t = table(test$adelie,knn.fit)
t
```

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

```r
confusionMatrix(knn.fit,reference = as.factor(test$adelie),positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 35  2
##          1 25 41
##
##                Accuracy : 0.7379
##                  95% CI : (0.642, 0.8196)
##     No Information Rate : 0.5825
##     P-Value [Acc > NIR] : 0.0007581
##
##                   Kappa : 0.499
##
##  Mcnemar's Test P-Value : 2.297e-05
##
##             Sensitivity : 0.9535
##             Specificity : 0.5833
##          Pos Pred Value : 0.6212
##          Neg Pred Value : 0.9459
##              Prevalence : 0.4175
##          Detection Rate : 0.3981
##    Detection Prevalence : 0.6408
##       Balanced Accuracy : 0.7684
##
##        'Positive' Class : 1
##
```

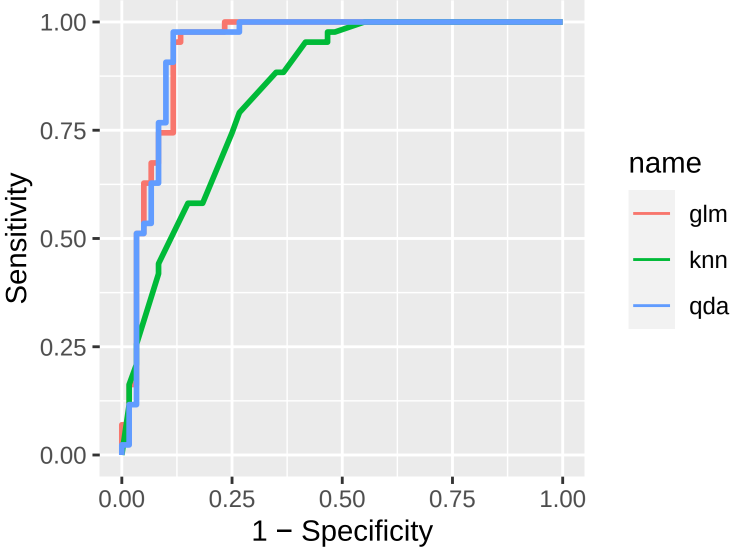
a)

iv)

Sensitivity and specificity are calculated in the function confusionMatrix(). But one can easily calculate them by hand using the formula $\frac{TP}{TP+TN}$ for sensitivity and $\frac{TN}{TN+FP}$ for specificity. We then get the following numbers. GLM: sensitivity = 0.9767, specificity = 0.8667, QDA: sensitivity = 0.9767, specificity = 0.7667, KNN: sensitivity = 0.9535, specificity = 0.5833.

```r
#ROC curves for the models at hand
probKNN = ifelse(knn.fit == 0, 1 - attributes(knn.fit)$prob, attributes(knn.fit)$prob)

log.Roc = roc(response = test$adelie,log.fit.p) #ROC for glm
qda.Roc = roc(response = test$adelie, qda.fit.p) #ROC for qda
```

```
knn.Roc = roc(response = test$adelie, probKNN) #ROC for knn

#constructing a data frame so it is easyy to plot all three cruves
#together in ggplot.

d.roc = data.frame(adelie = test$adelie, glm = log.fit.p, qda = qda.fit.p, knn = probKNN)

dl = melt_roc(d.roc, 'adelie', c('glm','qda','knn'))
ggplot(dl,aes(d=D,m=M,color=name)) + geom_roc(n.cuts = F) + xlab('1 - Specificity') + ylab('Sensitivity
```



```
auc(log.Roc) #Area under the curves, used to measure the fit of the model.
```

```
## Area under the curve: 0.9391
```

```
auc(qda.Roc)
```

```
## Area under the curve: 0.938
```

```
auc(knn.Roc)
```

```
## Area under the curve: 0.8417
```

b)

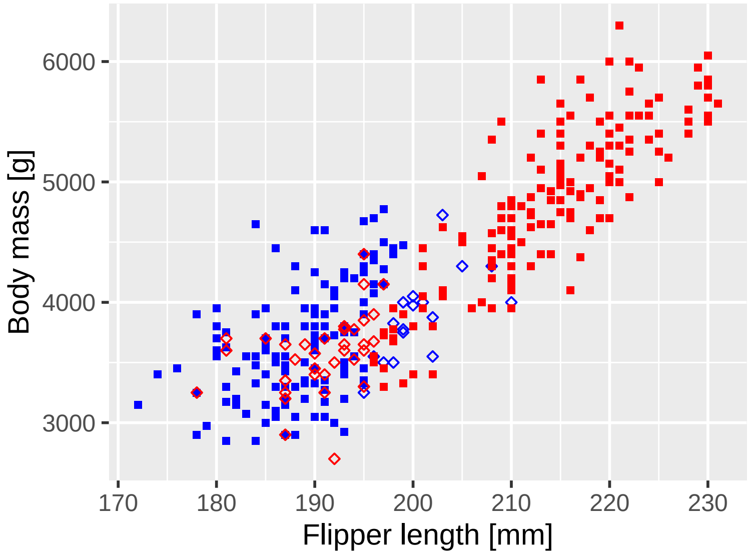
A ROC curve is a tool to help us build an understanding of how good a model is. We plot the sensitivity against 1 - specificity for all possible thresholds of the probability. From the ROC curves it can be observed that two the methods QDA and logistic regression are fairly equal when it comes to modeling if a penguin is Adelie or not, while KNN is slightly different. But given that the area under the curve is closest to one for the logistic regression, we can argue that this model performs best for this measure and therefore is our best model.

c)

Multiply by 2.038. Because eˆb*1000=2.038 so we multiply the odds by this number.

d)

```
#3d
#plot of the data
#We do similar to what we have done earlier for the two data set but
#now we use the whole data set penguins_reduced
log.adelie_prob = predict(log.fit, newdata = Penguins_reduced, type = 'response')
log.adelie_p = ifelse(log.adelie_prob > 0.5, yes = '1', no='0')

ggplot(data = Penguins_reduced, aes(flipper_length_mm, body_mass_g)) + geom_point(color=ifelse(Penguins_
             size =1.0,
             shape = ifelse(Penguins_reduced$adelie==log.adelie_p, 15,5))+
  xlab('Flipper length [mm]')+
  ylab('Body mass [g]')
```



Above is a plot of the classification of Adelie penguins. Blue = Adelie, Red = not Adelie, Square = correctly classified, Diamond = wrongly classified

Problem 4

a)

I) True
II) False
III) False
IV) False

b), c)

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

#logreg of the chd dataset
chd.log = glm(chd~sbp+sex+smoking,data = d.chd, family = 'binomial')
```

```
p0 = as.data.frame(t(c(150,1,0)))
colnames(p0) = c('sbp','sex','smoking')
probability = predict(chd.log, newdata = p0, type = 'response')
probability
```

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

```
#4c

B =1000
boot.fn = function(data,index){
  chd.log = glm(chd~sbp+sex+smoking,data = d.chd, subset = index, family = binomial())
  return(predict(chd.log,newdata = p0, type = 'response'))
}

p = rep(NA,B)
for (b in 1:B)
    p[b] = boot.fn(d.chd, sample(nrow(d.chd), size = nrow(d.chd), replace = T))
p.m = mean(p)
p.m
```

```
## [1] 0.1040857
```

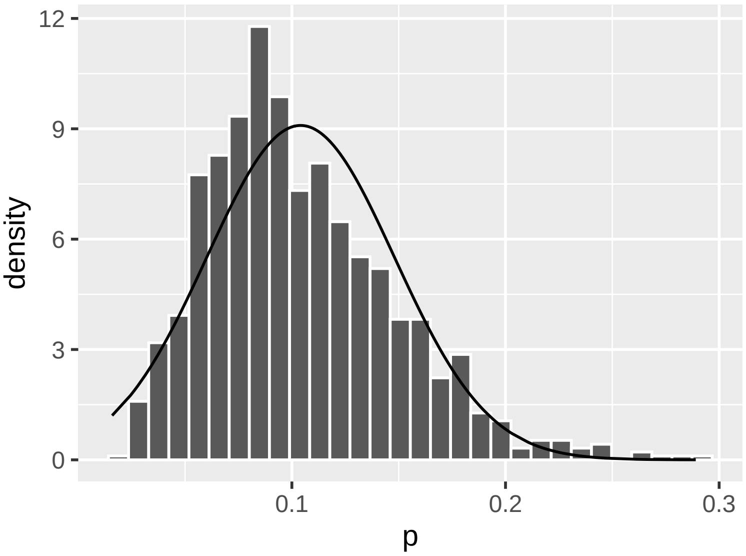
```
#standard deviation of p
sd(p)
```

```
## [1] 0.04388454
```

```
# 2.5% and 97.5% quantile
quantile(p, c(0.025,0.975))
```

```
##       2.5%      97.5%
## 0.03691636 0.20180577
```

```
#Plto of the distribution
distr = data.frame(p=p, norm_den = dnorm(p,mean(p), sd(p)))
ggplot(distr) +
  geom_histogram(aes(x=p, y = ..density..),color='white') +
  geom_line(aes(x=p, y = norm_den))
```



The probabilities from the bootstrap has an approximate standard deviation of 0.044. We can see that the expected probability is slightly above 0.1 for a non smoking man with systolic blood pressure of 150. The derivation of the 95% quantile interval tells us that with 95% certainty we can say that the probability of a man with the mentioned attributes having a coronary heart disease is approximately between 0.0369 and 0.202

d)

I) False
II) False
III) True
IV) True