Compulsory Exercise 3

TMA4268 Statistical Learning V2020

Martina Hall, Michail Spitieris, Stefanie Muff, Department of Mathematical Sciences, NTNU

Hand out date: April 14, 2020

Last changes: 13.04.2020

The submission deadline is Sunday, 3. May 2020, 23:59h using Blackboard

Introduction

Maximal score is 50 points. You need a score of 25/50 for the exercise to be approved. Your score will make up 50% points of your final grade.

Supervision

This project replaces the exam that you would have had to complete individually, thus we do not offer supervision as for compulsory 1 and 2. This avoids also that some students get advantages over others.

Practical issues

- **You work alone on this project.**
- **Your submitted pdf MUST NOT HAVE MORE THAN 14 pages! This is a requirement.** Any additional pages **will not be corrected**. Please only report the things that we ask for. For example, for single/multiple choice questions, only report the answers, as we do not grade your explanations in these cases.
- Remember to write your name on top of your submission.
- The exercise should be handed in as **one R Markdown file and a pdf-compiled version** of the R Markdown file (if you are not able to produce a pdf-file directly please make an html-file, open it in your browser and save as pdf - no, not landscape - but portrait please). We will read the pdf-file and use the Rmd file in case we need to check details in your submission.
- Please do not include all the text from this file (that you are reading now) - we want your R code, plots and written solutions - use the same template as for compulsory 1 (https://wiki.math.ntnu.no/tma4268/2020v/subpage6).
- Please save us time and do NOT submit word or zip, and do not submit only the Rmd. This only results in extra work for us!

R packages

You need to install the following packages in R to run the code in this file.

```r
install.packages("knitr")  #probably already installed
install.packages("rmarkdown")  #probably already installed
install.packages("ggplot2")  #plotting with ggplot
install.packages("ISLR")
intall.packages("boot")
install.packages("MASS")
install.packages("FactoMineR", dependencies = TRUE)
install.packages("factoextra")
install.packages("ggfortify")
install.packages("glmnet")
install.packages("tree")
install.packages("randomForest")
install.packages("gbm")
install.packages("ggfortify")
install.packages("keras")
install.packages("pls")
install.packages("gam")
```

Multiple/single choice problems

Some of the problems are *multiple choice* or *single choice questions*. This is how these will be graded:

- **Multiple choice questions (2P)**: There are four choices, and each of them can be TRUE or FALSE. If you make one mistake (either wrongly mark an option as TRUE/FALSE) you get 1P, if you have two or more mistakes, you get 0P. Your answer should be given as a list of answers, like TRUE, TRUE, FALSE, FALSE, for example.
- **Single choice questions (1P)**: There are four or five choices, and only *one* of the alternatives is the correct one. You will receive 1P if you choose the correct alternative and 0P if you choose wrong. Only say which option is true (for example (ii)).

Problem 1 (9P)

In compulsory exercise 2 we used the `College` data from the `ISLR` library, where we wanted to predict Outstate.

```r
library(ISLR)
library(keras)
set.seed(1)
College$Private = as.numeric(College$Private)
train.ind = sample(1:nrow(College), 0.5 * nrow(College))
college.train = College[train.ind, ]
college.test = College[-train.ind, ]
str(College)
```

```
## 'data.frame':    777 obs. of  18 variables:
##  $ Private    : num  2 2 2 2 2 2 2 2 2 2 ...
##  $ Apps       : num  1660 2186 1428 417 193 ...
##  $ Accept     : num  1232 1924 1097 349 146 ...
##  $ Enroll     : num  721 512 336 137 55 158 103 489 227 172 ...
##  $ Top10perc  : num  23 16 22 60 16 38 17 37 30 21 ...
##  $ Top25perc  : num  52 29 50 89 44 62 45 68 63 44 ...
##  $ F.Undergrad: num  2885 2683 1036 510 249 ...
##  $ P.Undergrad: num  537 1227 99 63 869 ...
##  $ Outstate   : num  7440 12280 11250 12960 7560 ...
##  $ Room.Board : num  3300 6450 3750 5450 4120 ...
##  $ Books      : num  450 750 400 450 800 500 500 450 300 660 ...
##  $ Personal   : num  2200 1500 1165 875 1500 ...
##  $ PhD        : num  70 29 53 92 76 67 90 89 79 40 ...
##  $ Terminal   : num  78 30 66 97 72 73 93 100 84 41 ...
##  $ S.F.Ratio  : num  18.1 12.2 12.9 7.7 11.9 9.4 11.5 13.7 11.3 11.5 ...
##  $ perc.alumni: num  12 16 30 37 2 11 26 37 23 15 ...
##  $ Expend     : num  7041 10527 8735 19016 10922 ...
##  $ Grad.Rate  : num  60 56 54 59 15 55 63 73 80 52 ...
```

The task here is to fit densely connected neural networks using the package `keras` in order to predict `Outstate`.

a) (2P)

Preprocessing is important before we fit a neural network. Apply feature-wise normalization to the predictors (but not to the response!).

Solution

```r
train.data = subset(college.train, select = -c(Outstate))
train.target = college.train$Outstate

test.data = subset(college.test, select = -c(Outstate))
test.target = college.test$Outstate

mean = apply(train.data, 2, mean)
std = apply(train.data, 2, sd)
train.data = scale(train.data, center = mean, scale = std)
test.data = scale(test.data, center = mean, scale = std)
```

Grading: If the test data were not scaled with the mean/std of the training data, -1P.

b) (2P)

Write down the equation which describes a network that predicts `Outstate` with 2 hidden layers and `relu` activation function with 64 units each. What activation function will you choose for the output layer?

Solution:

$$\hat{y}_1(x) = \gamma_{01} + \sum_{m=1}^{64} \gamma_{m1} \max(\beta_{0m} + \sum_{l=1}^{64} \beta_{lm} \max(\alpha_{0l} + \sum_{j=1}^{17} \alpha_{jl} x_j, 0), 0)$$

For the output layer we use linear (identity) activation function.

Grading: For each mistake or unclarity in the answer (e.g. if the activation functions are not explicitly specified) we subtract -1P.

c) (3P)

(i) Train the network from b) for the training data using the library `keras`; use 20% of the training data as your validation subset (1P).

(ii) Plot the training and validation error as a function of the epochs (1P).
(iii) Report the MSE of the test set and compare it with methods that you used in Compulsory 2 (1P).

Hints:

- Use the `optimizer = "rmsprop"` , `epochs=300` and `batch_size=8`
- Use `set.seed(123)` to make results comparable.
- Make sure that you are caching the results (`cache=TRUE` in the knitr options), because fitting the models takes some time, and you do not want to repeat this each time you compile your file.

Solution

Build Model

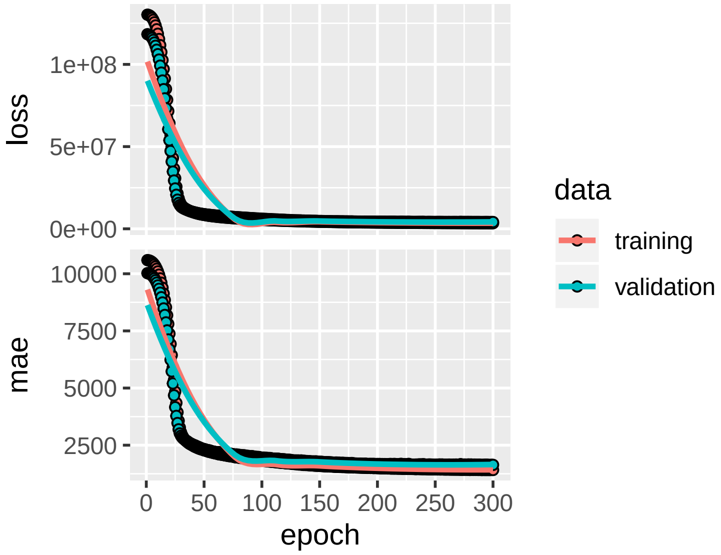
```
build_model = function() {
    model = keras_model_sequential() %>% layer_dense(units = 64, activation = "relu",
        input_shape = ncol(train.data)) %>% layer_dense(units = 64, activation = "relu") %>%
        layer_dense(units = 1)
    model %>% compile(optimizer = "rmsprop", loss = "mse", metrics = c("mae"))
}
```

Training the network

```
set.seed(123)

model <- build_model()
history = model %>% fit(train.data, train.target, epochs = 300, batch_size = 8,
    verbose = 0, validation_split = 0.2)

plot(history)
```



```
result = model %>% evaluate(test.data, test.target)
result$loss
```

```
## [1] 3693750
```

```
# str(history)
```

- It is important to mention that is the test mse is very sensitive to the seed.

d) (2P)

Apply one of the regularization techniques you heard about in the course (easiest to use dropout or weight decay with L1/L2 norms). Does this improve the performance of the network? Please again use `set.seed(123)` to make results comparable. Optional: You might try your own network architecture.

Solution:

1P for the right code to fit the network, 1P for the MSE and comparison to c).

Dropout

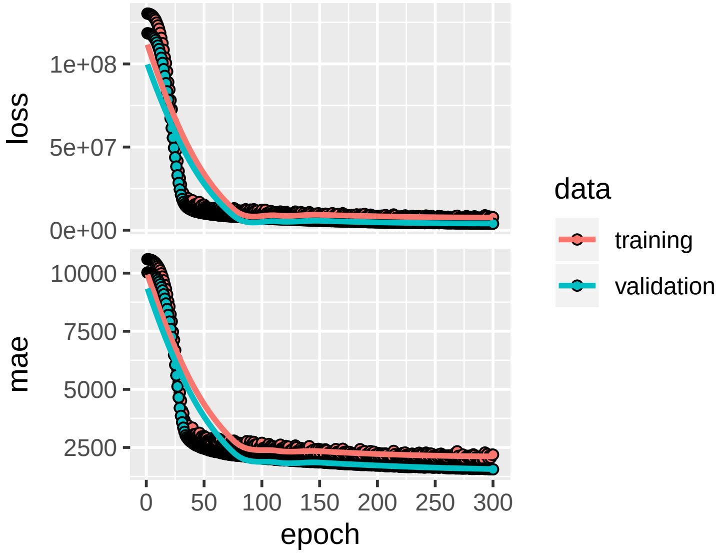
```
build_model = function() {
    model = keras_model_sequential() %>% layer_dense(units = 64, activation = "relu",
        input_shape = ncol(train.data)) %>% layer_dropout(rate = 0.4) %>% layer_dense(units = 64,
        activation = "relu") %>% layer_dropout(rate = 0.4) %>% layer_dense(units = 1)
    model %>% compile(optimizer = "rmsprop", loss = "mse", metrics = c("mae"))
}
```

```
# randomly allocate each sample to a particular fold
set.seed(123)

model_dropout = build_model()

history = model_dropout %>% fit(train.data, train.target, epochs = 300, batch_size = 8,
    verbose = 0, validation_split = 0.2)

plot(history)
```



```
result = model_dropout %>% evaluate(test.data, test.target)
result$loss
```

```
## [1] 3438301
```

L1 Regularization

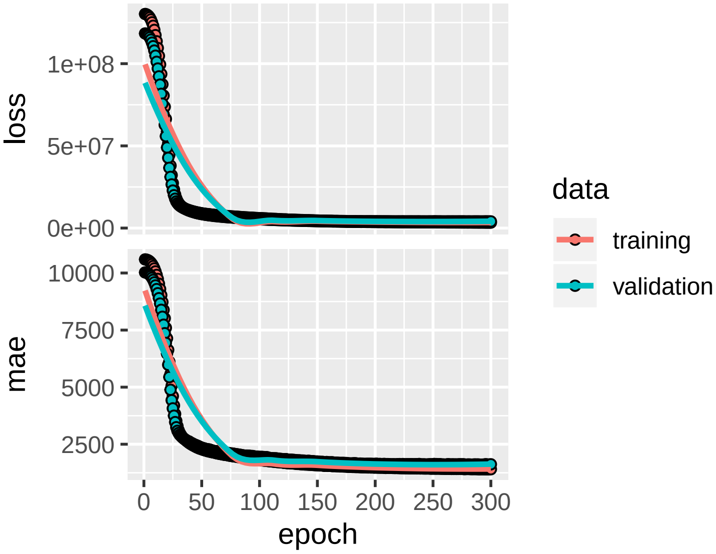
```
build_model = function() {
    model = keras_model_sequential() %>% layer_dense(units = 64, activation = "relu",
        input_shape = ncol(train.data), kernel_regularizer = regularizer_l2(l = 0.001)) %>%
        layer_dense(units = 64, activation = "relu", kernel_regularizer = regularizer_l2(l = 0.001)) %>%
        layer_dense(units = 1)
    model %>% compile(optimizer = "rmsprop", loss = "mse", metrics = c("mae"))
}

set.seed(123)

model_l1 = build_model()

history = model_l1 %>% fit(train.data, train.target, epochs = 300, batch_size = 8,
    verbose = 0, validation_split = 0.2)

plot(history)
```



```
result = model_l1 %>% evaluate(test.data, test.target)
result$loss
```

```
## [1] 3778037
```

- It seems that dropout improves the result from c, but keep in mind that there is some variability in the result. In any case, the test MSE does not get nearly as low as for the bagging, random forests and boosted regression trees.

Problem 2 (10P)

In this problem, we will use a real dataset of individuals with the Covid-19 infection. The data were downloaded from https://www.kaggle.com/shirmani/characteristics-corona-patients on 30. March 2020, and have only been cleaned for the purpose of this exercise. The dataset consists of 2010 individuals and four columns,

- `deceased`: if the person died of corona (1:yes, 0:no)

- sex: male/female
- age: age of person (ranging from 2 years to 99 years old)
- country: which country the person is from (France, Japan, Korea or Indonesia)

Note that the conclusions we will draw here are probably not scientifically valid, because we do not have enough information about how data were collected.

Load your data into R using the following code:

```
id <- "1CA1RPRYqU9oTIaHfSroitnWrI6WpUeBw"  # google file ID
d.corona <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    id), header = T)
```

a) Inspecting your data (1P)

Inspect the data by reporting **tables** for

- the number of deceased for each country,
- the number of deceased for each sex, and
- for each country: the number of deceased, separate for each sex.

Solution:

```
table(d.corona$deceased, d.corona$country)
```

```
##
##     France indonesia japan Korea
##   0    100        67   291  1507
##   1     14         2     3    26
```

```
table(d.corona$deceased, d.corona$sex)
```

```
##
##     female male
##   0   1075  890
##   1     14   31
```

```
France = d.corona[which(d.corona$country == "France"), ]
Japan = d.corona[which(d.corona$country == "japan"), ]
Indonesia = d.corona[which(d.corona$country == "indonesia"), ]
Korea = d.corona[which(d.corona$country == "Korea"), ]

table(France$deceased, France$sex)
```

```
##
##     female male
##   0     55   45
##   1      5    9
```

```
table(Japan$deceased, Japan$sex)
```

```
##
##     female male
##   0    120  171
##   1      0    3
```

```
table(Indonesia$deceased, Indonesia$sex)
```

```
##
##     female male
##   0     29   38
##   1      1    1
```

```
table(Korea$deceased, Korea$sex)
```

```
##
##     female male
##   0    871  636
##   1      8   18
```

b) Multiple choice (2P)

Answer the following multiple choice questions by using the data above to model the probability of deceased as a function of `sex`, `age` and `country` (with France as reference level; no interactions).

Which of the following statements are true, which false?

(i) Country is not a relevant variable in the model.

(ii) The slope for indonesia has a large p-value, which shows that we should remove the Indonesian population from the model, as they do not fit the model as well as the Japanese population.

(iii) Increasing the age by 10 years, $x_{age}^* = x_{age} + 10$, and holding all other covariates constant, the odds ratio to die increases by a factor of 1.97.

(iv) The probability to die is approximately 3.12 larger for males than for females.

Solution FALSE - FALSE - FALSE - FALSE

(iii) The calculation is $\exp(10 \cdot \beta_{age}) = 1.97$, but it's the odds that changes, not the odds ratio!
(iv) False, because $\exp(\beta_{sex})$ is the odds ratio, not the probability.

```
# Logistic regression
r.glm <- glm(deceased ~ sex + age + country, d.corona, family = "binomial")
summary(r.glm)
```

```
##
## Call:
## glm(formula = deceased ~ sex + age + country, family = "binomial",
##     data = d.corona)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2797  -0.1855  -0.1009  -0.0553   3.2233
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -7.633051   0.897063  -8.509  < 2e-16 ***
## sexmale            1.137246   0.343706   3.309 0.000937 ***
## age                0.068012   0.009846   6.907 4.94e-12 ***
## countryindonesia -0.754259   0.815127  -0.925 0.354796
## countryjapan     -2.434101   0.667826  -3.645 0.000268 ***
## countryKorea     -1.366797   0.374837  -3.646 0.000266 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 430.92  on 2009  degrees of freedom
## Residual deviance: 321.07  on 2004  degrees of freedom
## AIC: 333.07
##
## Number of Fisher Scoring iterations: 8
```

```r
anova(r.glm, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: deceased
##
## Terms added sequentially (first to last)
##
##
##         Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                     2009     430.92
## sex      1    9.969      2008     420.95 0.0015917 **
## age      1   80.567      2007     340.38 < 2.2e-16 ***
## country  3   19.313      2004     321.07 0.0002356 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

c) (1P)

Create a plot of probabilities to die of coronavirus as a function of age, separately for the two sexes and each country.

Hints:

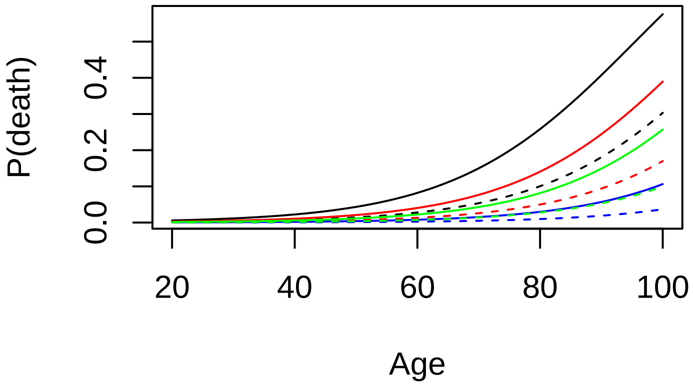
- Make one plot and add lines for each country/sex.
- A useful function to generate gridded data for prediction is `expand.grid()`. For example `newdata = expand.grid(sex="male",age= seq(20,100,1) ,country="France")` generates a grid for males in France over a range of ages between 20 and 100.

Solution:

```r
x.age <- seq(20, 100, 1)

yMaleFrance = predict(r.glm, newdata = expand.grid(sex = "male", age = x.age,
    country = "France"), type = "response")
yFemaleFrance = predict(r.glm, newdata = expand.grid(sex = "female", age = x.age,
    country = "France"), type = "response")
yMaleJapan = predict(r.glm, newdata = expand.grid(sex = "male", age = x.age,
    country = "japan"), type = "response")
yFemaleJapan = predict(r.glm, newdata = expand.grid(sex = "female", age = x.age,
    country = "japan"), type = "response")
yMaleIndonesia = predict(r.glm, newdata = expand.grid(sex = "male", age = x.age,
    country = "indonesia"), type = "response")
yFemaleIndonesia = predict(r.glm, newdata = expand.grid(sex = "female", age = x.age,
    country = "indonesia"), type = "response")
yMaleKorea = predict(r.glm, newdata = expand.grid(sex = "male", age = x.age,
    country = "Korea"), type = "response")
yFemaleKorea = predict(r.glm, newdata = expand.grid(sex = "female", age = x.age,
    country = "Korea"), type = "response")

plot(x = x.age, y = yMaleFrance, type = "l", ylab = "P(death)", xlab = "Age")
lines(x.age, yFemaleFrance, col = "black", lty = 2)
lines(x.age, yMaleJapan, col = "blue")
lines(x.age, yFemaleJapan, col = "blue", lty = 2)
lines(x.age, yMaleIndonesia, col = "red")
lines(x.age, yFemaleIndonesia, col = "red", lty = 2)
lines(x.age, yMaleKorea, col = "green")
lines(x.age, yFemaleKorea, col = "green", lty = 2)
```



d) (3P)

As a statistician working on these data, you are asked the following questions:

(i) Have males generally a higher probability to die of coronavirus than females?

(ii) Is age a greater risk factor for males than for females?

(iii) Is age a greater risk factor for the French population than for the Korean population?

Answer the questions by fitting appropriate models (1P each).

Solution:

Note that all the questions should be answered with full models, that is, using `age`, `sex` and `country` as predictors, because in b) we found that they were all important. Each time not all predictors were in the model we subtract -0.5P. In (ii) and (iii) points can only be gained if interaction terms appear in the models.

(i) **Yes**, this is clearly the case, as can be seen from the model fitted under b).

(ii) **No**. The slope for women is β_2, while the slope for males is $\beta_2 + \beta_6$. As we have no evidence that β_6 is different from zero ($p = 0.89$), we have no evidence that age is a greater risk for males than for females.

```
mod2 = glm(deceased ~ sex + age + country + sex * age, d.corona, family = "binomial")
summary(mod2)
```

```
##
## Call:
## glm(formula = deceased ~ sex + age + country + sex * age, family = "binomial",
##     data = d.corona)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -1.2701  -0.1861  -0.1028  -0.0563   3.2162
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -7.77989    1.41136  -5.512 3.54e-08 ***
## sexmale           1.35802    1.65509   0.821 0.411924
## age               0.06986    0.01683   4.150 3.32e-05 ***
## countryindonesia -0.75872    0.81526  -0.931 0.352035
## countryjapan     -2.43169    0.66785  -3.641 0.000271 ***
## countryKorea     -1.36693    0.37461  -3.649 0.000263 ***
## sexmale:age      -0.00282    0.02064  -0.137 0.891338
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 430.92  on 2009  degrees of freedom
## Residual deviance: 321.05  on 2003  degrees of freedom
## AIC: 335.05
##
## Number of Fisher Scoring iterations: 8
```

(iii) **No**. The slope for age for the Korean population is $\beta_2 + \beta_8$, while it is β_2 for French. Given that $p = 0.35$ for β_8, so there is no evidence that age is an increased risk for French with respect to Korean.

```
mod3 = glm(deceased ~ sex + age + country + age * country, d.corona, family = "binomial")
summary(mod3)
```

```
##
## Call:
## glm(formula = deceased ~ sex + age + country + age * country,
##     family = "binomial", data = d.corona)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.53644  -0.17938  -0.10094  -0.05524   3.16223
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -10.35138    2.52507  -4.099 4.14e-05 ***
## sexmale                1.17341    0.34949   3.357 0.000787 ***
## age                    0.10092    0.02969   3.399 0.000676 ***
## countryindonesia       5.66073    3.33056   1.700 0.089200 .
## countryjapan           3.35895    3.31974   1.012 0.311628
## countryKorea           1.09736    2.66139   0.412 0.680100
## age:countryindonesia  -0.09372    0.04974  -1.884 0.059573 .
## age:countryjapan      -0.07559    0.04343  -1.741 0.081755 .
## age:countryKorea      -0.02991    0.03193  -0.937 0.348933
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 430.92  on 2009  degrees of freedom
## Residual deviance: 315.74  on 2001  degrees of freedom
## AIC: 333.74
##
## Number of Fisher Scoring iterations: 8
```

e) Interpret your model (1P)

According to your model fitted in part b), it looks like the French population is at a much higher risk of dying from Covid-19 than the other countries. Do you trust this result? How could it be influenced by the way the data were collected?

Solution:

It will probably have to do with how it was tested, that is, in France only severe cases were tested, whereas asian countries have tested much more.

f) Multiple choice (2P)

Which of the following statements are true, which false?

Consider the classification tree below to answer:

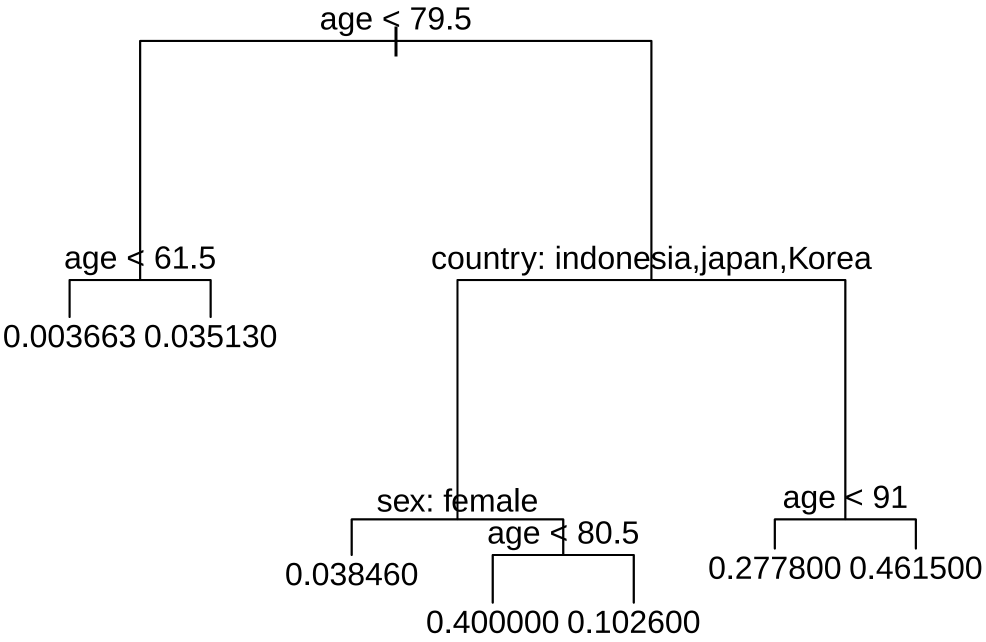
(i) The probability of dying (`deceased = 1`) is about 0.46 for a French person with age above 91.

(ii) Age seems to be a more important predictor for mortality than sex.

Consider the LDA code and output below:

(iii) The "null rate" for misclassification is 2.24%, because this is the proportion of deaths among all cases in the dataset. No classifier should have a higher misclassification rate.

(iv) LDA is not a very useful method for this dataset, among other reasons because it does not estimate probabilities, but also because the misclassification error is too high.



```r
library(MASS)
table(predict = predict(lda(deceased ~ age + sex + country, data = d.corona))$class,
    true = d.corona$deceased)
```

```
##        true
## predict    0    1
##       0 1926   31
##       1   39   14
```

Solution:

TRUE, TRUE, TRUE/FALSE, TRUE/FALSE

Note: (iii) and (iv) were later found to be ambiguous. Therefore T and F were graded as “correct”.

Problem 3 (14P)

The d.support dataset (source *F. E. Harrell, Regression Modeling Strategies*) contains the total hospital costs of 9105 patients with certain diseases in American hospitals between 1989 and 1991. The different variables are

Variable	Meaning
totcst	Total costs
age	Age of the patients
‘dzgroup	‘ Disease group
num.co	Number of co-morbidities
edu	Years of education
scoma	Measure for Glasgow coma scale
income	Income
race	Rasse
meanbp	Mean blood pressure
hrt	Heart rate
resp	Respiratory frequency
temp	Body temperature
pafi	PaO2/FiO2 proportion (blood-gas mixture)

Data are loaded as follows (and we reduce the number of patients to the 4960 complete cases with total costs larger than 0):

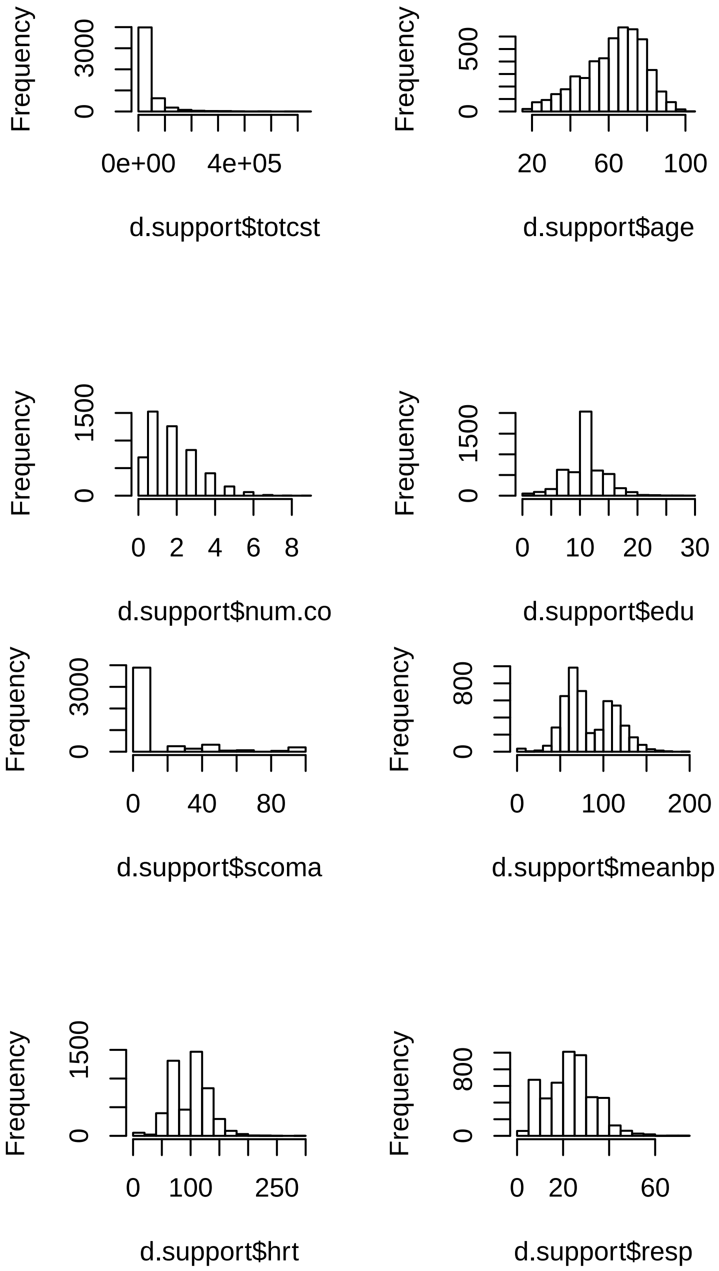
```r
id <- "1heRtzi8vBoBGMaM2-ivBQI5Ki3HgJTmO"  # google file ID
d.support <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    id), header = T)
# We only look at complete cases
d.support <- d.support[complete.cases(d.support), ]
d.support <- d.support[d.support$totcst > 0, ]
```

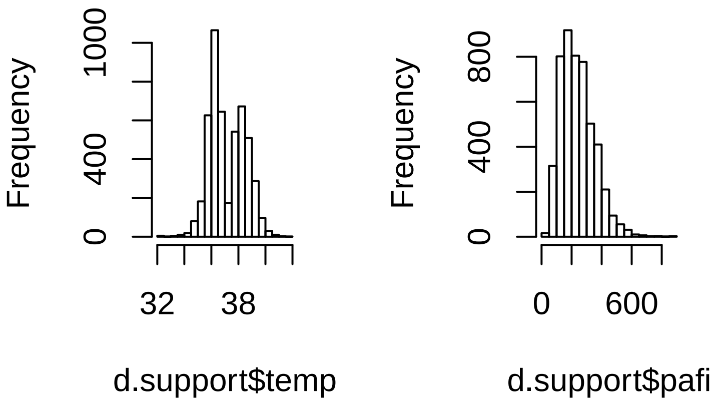
We would like to build models that help us to understand which predictors are mostly driving the total cost, but also models for prediction.

a) (1P)

Before we start analysing the data, visualize the distributions of all continuous or integer variables with histograms. Suggest a transformation for the response variable totcst (hint: it is a *standard transformation* that we have used earlier in the course). Important: **you should fit all models with the transformed version of the response variable `totcst` from now on. Leave all other variables untransformed.**

Solution:





The distribution of the total costs is extremely right-skewed. We will thus use `log(totcst)` for the rest of the problem.

b) (3P)

Fit a multiple linear regression model with the six covariates `age`, `temp`, `edu`, `resp`, `num.co` and `dzgroup` and the (transformed version of the) response `totcst`.

(i) How much /by which factor are the total costs expected to change when a patient's age increases by 10 years, given that all other characteristica of the patient are the same? Use the transformed respose to fit the model, but report the result on the original (back-transformed) scale of the response. (1P)
(ii) Do a residual analysis using the Tukey-Anscombe plot and the QQ-diagram. Are the assumptions fulfilled? (1P)
(iii) Does the effect of age depend on the disease group? Do a formal test and report the p-value. (1P)

Solution:

```
r.lm <- lm(log(totcst) ~ age + temp + edu + resp + num.co + dzgroup, d.support)
summary(r.lm)
```

```
##
## Call:
## lm(formula = log(totcst) ~ age + temp + edu + resp + num.co +
##     dzgroup, data = d.support)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.6554 -0.6524 -0.0437  0.6203  3.5226
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             8.0823597  0.4014491  20.133  < 2e-16 ***
## age                    -0.0069950  0.0008742  -8.001 1.52e-15 ***
## temp                    0.0690123  0.0104548   6.601 4.51e-11 ***
## edu                     0.0249934  0.0039506   6.326 2.73e-10 ***
## resp                   -0.0027792  0.0012791  -2.173   0.0298 *
## num.co                 -0.0430856  0.0107460  -4.009 6.18e-05 ***
## dzgroupCHF             -1.3992569  0.0437688 -31.969  < 2e-16 ***
## dzgroupCirrhosis       -0.9113548  0.0645311 -14.123  < 2e-16 ***
## dzgroupColon Cancer    -1.4947386  0.0842719 -17.737  < 2e-16 ***
## dzgroupComa            -0.4501610  0.0562858  -7.998 1.57e-15 ***
## dzgroupCOPD            -1.2432540  0.0441240 -28.176  < 2e-16 ***
## dzgroupLung Cancer  -1.6924699  0.0540838 -31.293  < 2e-16 ***
## dzgroupMOSF w/Malig -0.2627110  0.0510358  -5.148 2.74e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9337 on 4947 degrees of freedom
## Multiple R-squared:  0.3817, Adjusted R-squared:  0.3802
## F-statistic: 254.5 on 12 and 4947 DF,  p-value: < 2.2e-16
```

```
# anova(r.lm)

r.lm2 <- lm(log(totcst) ~ age + temp + edu + resp + num.co + dzgroup * age,
    d.support)
summary(r.lm2)
```

```
##
## Call:
## lm(formula = log(totcst) ~ age + temp + edu + resp + num.co +
##     dzgroup * age, data = d.support)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.6560 -0.6571 -0.0379  0.6180  3.5006
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                7.916235   0.405255  19.534  < 2e-16 ***
## age                       -0.004773   0.001144  -4.170 3.09e-05 ***
## temp                       0.069829   0.010450   6.682 2.61e-11 ***
## edu                        0.024749   0.003945   6.274 3.82e-10 ***
## resp                      -0.002709   0.001279  -2.118 0.034196 *
## num.co                    -0.041790   0.010740  -3.891 0.000101 ***
## dzgroupCHF                -1.038131   0.188132  -5.518 3.60e-08 ***
## dzgroupCirrhosis          -0.554501   0.257563  -2.153 0.031377 *
## dzgroupColon Cancer       -2.024012   0.536676  -3.771 0.000164 ***
## dzgroupComa                0.267332   0.216580   1.234 0.217137
## dzgroupCOPD               -1.466316   0.252269  -5.813 6.54e-09 ***
## dzgroupLung Cancer        -1.913060   0.297221  -6.436 1.34e-10 ***
## dzgroupMOSF w/Malig        0.363018   0.197146   1.841 0.065628 .
## age:dzgroupCHF            -0.005603   0.002761  -2.029 0.042516 *
## age:dzgroupCirrhosis      -0.006396   0.004624  -1.383 0.166608
## age:dzgroupColon Cancer    0.008119   0.008254   0.984 0.325305
## age:dzgroupComa           -0.011423   0.003322  -3.439 0.000589 ***
## age:dzgroupCOPD            0.002905   0.003606   0.805 0.420610
## age:dzgroupLung Cancer     0.003533   0.004734   0.746 0.455533
## age:dzgroupMOSF w/Malig -0.010539   0.003215  -3.278 0.001053 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9317 on 4940 degrees of freedom
## Multiple R-squared:  0.3852, Adjusted R-squared:  0.3828
## F-statistic: 162.9 on 19 and 4940 DF,  p-value: < 2.2e-16
```

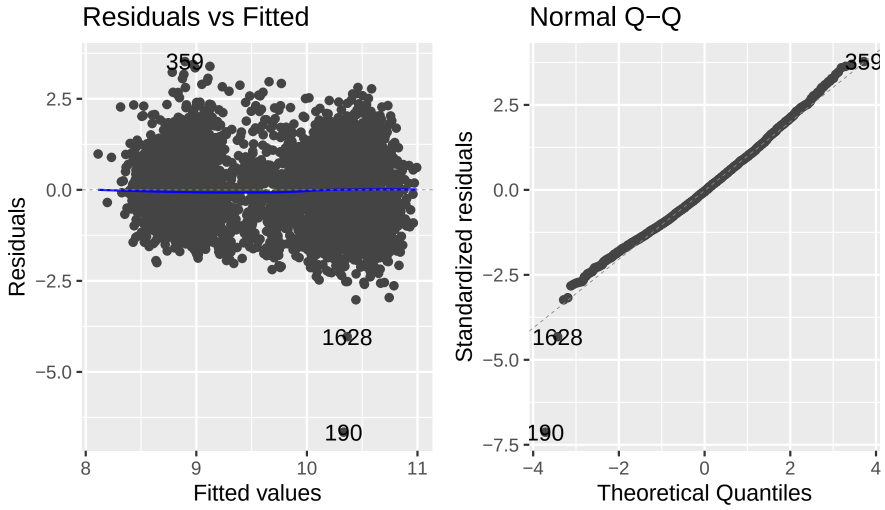
```r
anova(r.lm2)
```

```
## Analysis of Variance Table
##
## Response: log(totcst)
##               Df Sum Sq Mean Sq  F value    Pr(>F)
## age            1  279.6 279.590 322.0778 < 2.2e-16 ***
## temp           1  195.0 194.988 224.6195 < 2.2e-16 ***
## edu            1   68.2  68.207  78.5718 < 2.2e-16 ***
## resp           1    6.5   6.540   7.5339 0.0060768 **
## num.co         1  268.9 268.942 309.8116 < 2.2e-16 ***
## dzgroup        7 1844.0 263.431 303.4637 < 2.2e-16 ***
## age:dzgroup    7   24.5   3.506   4.0387 0.0002019 ***
## Residuals   4940 4288.3   0.868
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(i) The expected costs should go down by a factor of exp(10· (-0.007)) = 0.932. Note that we need the exp-transformation to obtain the result on the scale of the costs (instead of log costs, which we used as the response).

(ii)

```r
library(ggfortify)
autoplot(r.lm, which = c(1, 2))
```



No violations visible.

(iii) Yes, in the anova table we see that the p-value for the interaction between age and disease group is about $p = 0.0002$, thus very small. The full point was only given if all the covariates and the interaction term were in the model, and the p-value was used to draw the appropriate conclusion.

c) (3P)

In order to build a more robust model for inference and prediction of the total costs, continue using ridge regression. Create a training set with 80% of the data and a test set with the remaining 20% (1P). Run cross-validation to find the largest value of λ such that the error is within 1 standard error of the smallest λ (1P). Report the test MSE of the ridge regression where you used the respective λ (1P).

Be careful: we still use the same transformation for the response as in b) – you should report the MSE using the transformed version of `totcst` (i.e., do **not back-transform** the MSE to the original scale)

R-hints:

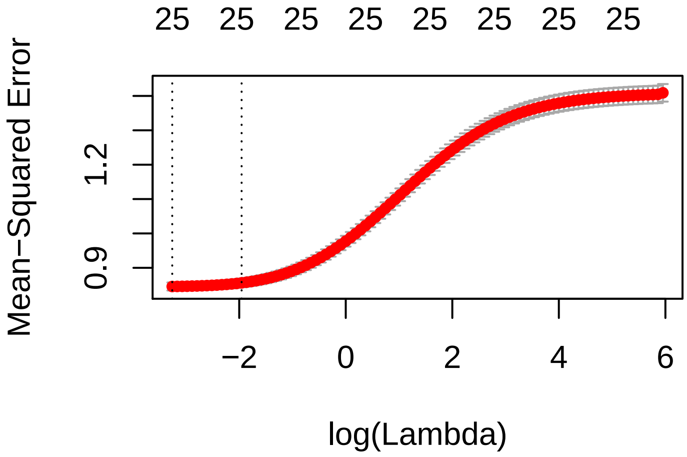
```
set.seed(12345)
train.ind = sample(1:nrow(d.support), ... * nrow(d.support))
d.support.train = d.support[..., ]
d.support.test = d.support[..., ]
```

Solution:

We recommend to use all covariats (not just those used above), but no points were subtracted if someone went on with the variables from the model in b).

```
set.seed(12345)
train.ind = sample(1:nrow(d.support), 0.8 * nrow(d.support))
d.support.train = d.support[train.ind, ]
d.support.test = d.support[-train.ind, ]
```

```
library(glmnet)
x.train = model.matrix(log(totcst) ~ ., data = d.support.train)[, -1]
y.train = log(d.support.train$totcst)
set.seed(4268)
cv.ridge = cv.glmnet(x.train, y.train, alpha = 0)
plot(cv.ridge)
```



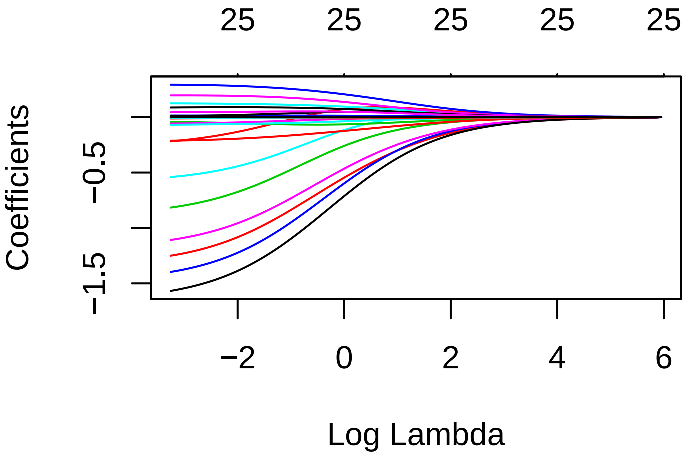
```
(lamda.ridge = cv.ridge$lambda.1se)
```

```
## [1] 0.1415939
```

```
r.ridge = glmnet(x.train, y.train, alpha = 0, lambda = cv.ridge$lambda.1se)
coef(r.ridge)
```

```
## 26 x 1 sparse Matrix of class "dgCMatrix"
##                                s0
## (Intercept)          8.7509876866
## age                 -0.0063660700
## dzgroupCHF          -1.0745066758
## dzgroupCirrhosis    -0.6676111932
## dzgroupColon Cancer -1.2144632825
## dzgroupComa         -0.4382669579
## dzgroupCOPD         -0.9492741087
## dzgroupLung Cancer  -1.3775286478
## dzgroupMOSF w/Malig -0.1282084246
## num.co              -0.0548536914
## edu                  0.0145426381
## income>$50k          0.1197607346
## income$11-$25k      -0.0467792620
## income$25-$50k       0.0205420134
## incomeunder $11k    -0.1932619994
## scoma                0.0034426327
## raceasian            0.2822436227
## raceblack           -0.0598678518
## racehispanic         0.1926428042
## raceother            0.0908544458
## racewhite           -0.0080991686
## meanbp               0.0002068517
## hrt                  0.0028450861
## resp                -0.0048272561
## temp                 0.0480237414
## pafi                -0.0005729920
```

```
plot(glmnet(x.train, y.train, alpha = 0), "lambda")
```



```
x.test = model.matrix(log(totcst) ~ ., data = d.support.test)[, -1]
y.test = d.support.test$totcst
mse.ridge = mean((log(y.test) - predict(r.ridge, newx = x.test))^2)
mse.ridge
```

```
## [1] 0.874485
```

d) (3P)

Now assume that our sole aim is prediction. In the course you heard about *partial least squares (PLS)*. It is a smart approach that uses the principal component regression idea, but finds the components that are best correlated with the response.

Proceed as follows:

(i) Run a PLS regression (don't forget to scale the variables, `scale=TRUE`) (1P).
(ii) Choose an optimal number of principal components (PCs) using cross-validation (1P).

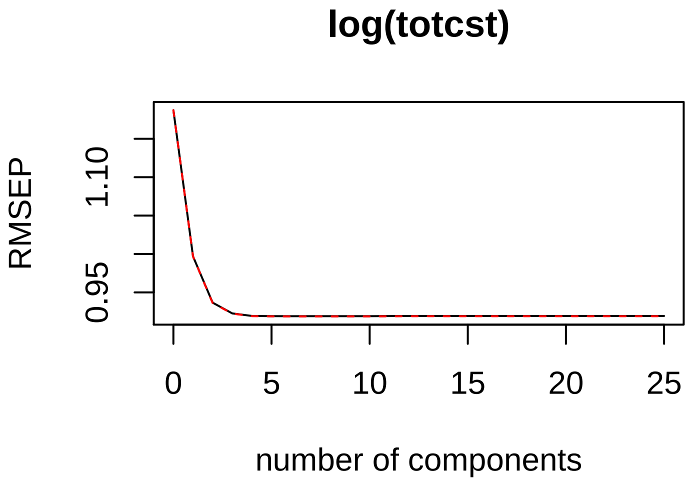
(iii) Report the MSE of the test set when using the respective set of PCs and compare to the result from ridge regression. Conclusion? (1P)

Solution:

```
library(pls)
set.seed(234)
r.pls <- plsr(log(totcst) ~ ., data = d.support.train, scale = T, validation = "CV")
summary(r.pls)
```

```
## Data:    X dimension: 3968 25
##  Y dimension: 3968 1
## Fit method: kernelpls
## Number of components considered: 25
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           1.187   0.9969   0.9366   0.9226   0.9193   0.9191   0.9190
## adjCV        1.187   0.9967   0.9361   0.9223   0.9190   0.9188   0.9187
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV      0.9190   0.9190   0.9190    0.9190    0.9192    0.9193    0.9193
## adjCV   0.9187   0.9187   0.9187    0.9187    0.9189    0.9190    0.9190
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
## CV       0.9193    0.9193    0.9193    0.9193    0.9193    0.9193
## adjCV    0.9190    0.9190    0.9190    0.9190    0.9190    0.9190
##        20 comps  21 comps  22 comps  23 comps  24 comps  25 comps
## CV       0.9193    0.9193    0.9193    0.9193    0.9193    0.9193
## adjCV    0.9190    0.9190    0.9190    0.9190    0.9190    0.9190
##
## TRAINING: % variance explained
##              1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X              7.938    11.79    16.86    22.81    27.30    31.57    35.05
## log(totcst)   29.947    38.61    40.30    40.69    40.74    40.75    40.75
##              8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## X              38.94    43.78     47.74     50.34     52.94     55.23
## log(totcst)    40.75    40.75     40.75     40.75     40.75     40.75
##              14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
## X               57.47     61.74     65.10     68.77     72.02     76.31
## log(totcst)     40.75     40.75     40.75     40.75     40.75     40.75
##              20 comps  21 comps  22 comps  23 comps  24 comps  25 comps
## X               80.17     83.53     87.97     92.00     96.01    100.00
## log(totcst)     40.75     40.75     40.75     40.75     40.75     40.75
```

```
validationplot(r.pls)
```



```
# 6 or 7 components is the smallest number for several same CV errors
mse.pls = mean((log(d.support.test$totcst) - predict(r.pls, x.test, ncomp = 6))^2)
mse.pls
```

```
## [1] 0.8624827
```

4 to 7 components were seen as ok. Any number above gave -1P, because more only increase complexity without any gain.

It looks like by using 6 or 7 PCs in PLS we can get a somewhat lower prediction error than for ridge regression. PLS is thus the preferred method, although the difference is small.

e) (4P)

Now choose two other methods that you know from the course and try to build models with even lower test MSEs than those found so far (imagine that this is a competition where the lowest test MSE wins). Use the same training and test dataset as generated above. And remember that we are still *always* working with the transformed version of the response variable (`totcst`). In particular, use

(i) One model that involves non-linear transformations of the covariates (e.g., splines, natural splines, polynomials etc) that are combined to a GAM (2P).
(ii) One model/method based on regression trees (2P).

Very briefly discuss or explain your choices (1-2 sentences each).

Solution:

Here we are basically checking of what the student did is theoretically sound and correct, and then we check that the test MSEs were indeed smaller that those found above. A possible GAM solution is given as follows:

```
library(gam)
gam.fit = gam(log(totcst) ~ poly(age, degree = 2) + dzgroup + num.co + s(edu,
    df = 3) + income + poly(scoma, degree = 2) + race + ns(meanbp, df = 4) +
    ns(hrt, df = 4) + ns(resp, df = 4) + ns(temp, df = 4), data = d.support.train)

pred = predict(gam.fit, d.support.test)

mse.gam = mean((pred - log(d.support.test$totcst))^2)
mse.gam
```

```
## [1] 0.8291008
```

And boosting fit hyperparameter chosen as follows wins the challenge:

```
set.seed(4268)
library(gbm)
boost = gbm(log(totcst) ~ ., d.support.train, distribution = "gaussian", n.trees = 1000,
    interaction.depth = 4, shrinkage = 0.01)
yhat.boost = predict(boost, newdata = d.support.test, n.trees = 1000)
mse.boost = mean((log(d.support.test$totcst) - yhat.boost)^2)
mse.boost
```

```
## [1] 0.8201154
```

Problem 4 (Mixed questions; 6P)

a) 2P

We look at the following cubic regression spline model:

$$Y = \begin{cases} \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \epsilon \ , & \text{if } \ x \leq 1 \ , \\ \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x-1)^3 + \epsilon \ , & \text{if } \ 1 < x \leq 2 \ , \\ \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x-1)^3 + \beta_5 (x-2)^3 + \epsilon \ , & \text{if } \ x > 2 \ . \end{cases}$$

Write down the basis functions (1P) and the design matrix (1P) of this model.

Solution:

Basis functions: $b_1(x) = x$, $b_2(x) = x^2$, $b_3(x) = x^3$, $b_4(x) = (x-1)^3_+$, and $b_5(x) = (x-2)^3_+$.

Design matrix for n data points:

$$\mathbf{X} = \begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 & (x_1-1)^3_+ & (x_1-2)^3_+ \\ 1 & x_2 & x_2^2 & x_2^3 & (x_2-1)^3_+ & (x_2-2)^3_+ \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 & (x_n-1)^3_+ & (x_n-2)^3_+ \end{pmatrix} \ .$$

b) Multiple choice - 2P

Inference vs prediction: Which of the following methods are suitable when the aim of your analysis is inference?

(i) Lasso and ridge regression
(ii) Multiple linear regression with interaction terms
(iii) Logistic regression
(iv) Support Vector Machines

Solution TRUE - TRUE - TRUE - FALSE

c) Multiple choice - 2P

We again look at the Covid-19 dataset from Problem 2 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`, `age` and `country` 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.

Look at the R output below and compare the standard errors that we obtain from these two approaches (note that the `t1*` to `t6*` variables are sorted in the same way as for the `glm()` output).

```r
library(boot)
boot.fn <- function(data, index) {
    return(coefficients(glm(deceased ~ sex + age + country, family = "binomial",
        data = data, subset = index)))
}
boot(d.corona, boot.fn, 1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = d.corona, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##        original       bias    std. error
## t1* -7.63305130 -0.116372492 0.759199393
## t2*  1.13724644  0.019625033 0.376706321
## t3*  0.06801169  0.000586064 0.007925872
## t4* -0.75425940 -2.116500078 5.380853133
## t5* -2.43410057 -0.685377375 3.100291962
## t6* -1.36679680  0.013719644 0.387964250
```

```r
# Logistic regression
r.glm <- glm(deceased ~ sex + age + country, d.corona, family = "binomial")
summary(r.glm)$coef
```

```
##                      Estimate  Std. Error    z value     Pr(>|z|)
## (Intercept)       -7.63305130 0.897063042 -8.5089352 1.755379e-17
## sexmale            1.13724644 0.343705727  3.3087794 9.370363e-04
## age                0.06801169 0.009846377  6.9072806 4.940322e-12
## countryindonesia -0.75425940 0.815127165 -0.9253273 3.547957e-01
## countryjapan      -2.43410057 0.667826265 -3.6448111 2.675883e-04
## countryKorea      -1.36679680 0.374836917 -3.6463772 2.659635e-04
```

Which of the following statements are true?

(i) There are large differences between the estimated standard errors, which indicates a problem with the bootstrap.
(ii) The differences between the estimated standard errors indicate a problem with the assumptions taken about the distribution of the estimated parameters in logistic regression.
(iii) The `glm` function leads to too small p-values for the differences between countries, in particular for the differences between Indonesia and France and between Japan and France.
(iv) The bootstrap relies on random sampling the same data without replacement.

Solution:

FALSE - TRUE - TRUE - FALSE

(iv) it would be with replacement

Problem 5 (Multiple and single choice questions; 11P)

a) Multiple choice - 2P

Which of the following are techniques for regularization?

(i) Lasso
(ii) Ridge regression
(iii) Forward and backward selection
(iv) Stochastic gradient descent

Solution: TRUE - TRUE - FALSE - TRUE

b) Multiple choice - 2P

Which of the following statements about principal component regression (PCR) and partial least squares (PLS) are correct?

(i) PCR involves the first principal components that are most correlated with the response.
(ii) PLS involves the first principal components that are most correlated with the response.
(iii) The idea in PLS is that we choose the principal components that explain most variation among all covariates.
(iv) The idea in PCR is that we choose the principal components that explain most variation among all covariates.

Solution FALSE - TRUE - FALSE - TRUE

c) Single choice - 1P

In ridge regression, we estimate the regression coefficients in a linear regression model by minimizing

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j-1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 \ .$$

What happens when we increase λ from 0? Choose the single correct statement:

(i) The training RSS will steadily decrease.
(ii) The test RSS will steadily decrease.
(iii) The test RSS will steadily increase.
(iv) The bias will steadily increase.
(v) The variance of the estimator will steadily increase.

Solution: (iv) is correct.

d) Single choice - 1P

Which statement about the *curse of dimensionality* is correct?

(i) It means that we have a bias-variance tradeoff in K-nearest neighbor regression, where large K leads to more bias but less variance for the predictor function.
(ii) It means that the performance of the K-nearest neighbor classifier gets worse when the number of predictor variables p is large.

(iii) It means that the K-means clustering algorithm performs bad if the datapoints lie in a high-dimensional space.
(iv) It means that support vector machines with radial kernel function should be avoided, because radial kernels correspond to infinite-dimensional polynomial boundaries.
(v) It means that we should never measure too many covariates when we want to do classification.

Solution: (ii) is correct.

e) Single choice - 1P

Now assume you have 10 covariates, X_1 to X_{10}, each of them uniformly distributed in the interval $[0, 1]$. To predict a new test observation $(X_1^{(0)}, \ldots, X_{10}^{(0)})$ in a K-nearest neighbor (KNN) clustering approach, we use all observations within 20% of the range closest to each of the covariates (that is, in each dimension). Which proportion of available (training) observations can you expect to use for prediction?

(i) $1.02 \cdot 10^{-7}$
(ii) $2.0 \cdot 10^{-3}$
(iii) 0.20
(iv) 0.04
(v) 10^{-10}

Solution: (i) is correct.

f) Multiple choice - 2P

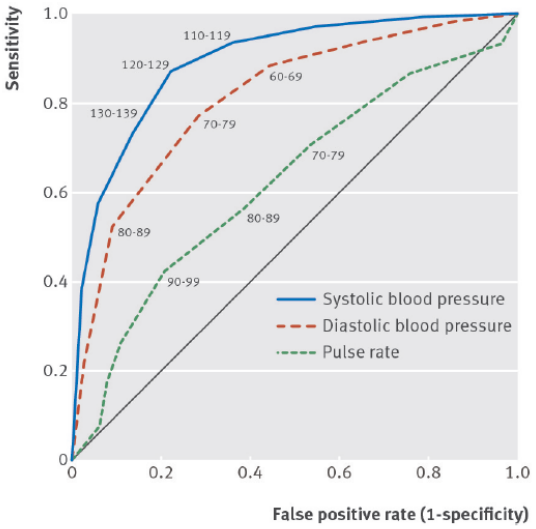
This example is taken from a real clinical study by *Ikeda, Matsunaga, Irabu, et al. Using vital signs to diagnose impaired consciousness: cross sectional observational study. BMJ 2002;325:800.* Researchers investigated the use of vital signs as a screening test to identify brain lesions in patients with impaired consciousness. The setting was an emergency department in Japan. The study included 529 consecutive patients that arrived with consciousness. Patients were followed until discharge. The vital signs of systolic and diastolic blood pressure and pulse rate were recorded on arrival. The aim of this study was to find a quick test for assessing whether the newly arrived patient suffered from a brain lesion. While vital signs can be measured immediately, the actual diagnosis of a brain lesion can only be determined on the basis of brain imaging and neurological examination at a later stage, thus the quick measurements of blood pressure and heart rate are important to make a quick assessment. In total, 312 patients (59%) were diagnosed with a brain lesion.

The performance of each vital sign (systolic blood pressure, diastolic blood pressure and heart rate) was separately evaluated as a screening test to quickly diagnose brain lesions. To assess the quality of each of these vital signs, different thresholds were taken successively to discriminate between “negative” and “positive” screening test result. For each vital sign and each threshold the sensitivity and specificity were derived and used to plot a receiver operating characteristic (ROC) curve for the vital sign (Figure 1):

Which of the following statements are true?

(i) The value of 1-specificity represents the proportion of patients without a diagnosed brain lesion identified as positive on screening.
(ii) When we use different cut-offs, sensitivity increases at the cost of lower specificity, and vice versa.
(iii) A perfect diagnostic test has an AUC of 0.5.
(iv) The vital sign that is most suitable to distinguish between patients with and without brain lesion is systolic blood pressure.

Solution: TRUE - TRUE - FALSE - TRUE



Receiver operating characteristic curves for each of the three vital signs as screening tests for diagnosed brain lesions. For each vital sign, selected cut-off points between a positive and negative screening test result are shown

Figure 1: Figure for problem 5f); taken from *P. Sedgwick, BMJ 2011;343*

g) Multiple choice - 2P

We study the `decathlon2` dataset from the `factoextra` package in R, where Athletes' performance during a sporting meeting was recorded. We look at 23 athletes and the results from the 10 disciplines in two competitions. Some rows of the dataset are displayed here:

```
decathlon2.active[c(1, 3, 4), ]
```

```
##          100m long_jump shot_put high_jump  400m 110.hurdle discus
## SEBRLE  11.04      7.58    14.83      2.07 49.81      14.69  43.75
## BERNARD 11.02      7.23    14.25      1.92 48.93      14.99  40.87
## YURKOV  11.34      7.09    15.19      2.10 50.42      15.31  46.26
##         pole_vault javeline 1500m
## SEBRLE        5.02    63.19 291.7
## BERNARD       5.32    62.77 280.1
## YURKOV        4.72    63.44 276.4
```

From a principal component analysis we obtain the biplot given in Figure 2.

Which of the following statements are true, which false?

(i) The athlete named CLAY seems to be one of the fastest 1500m runners.
(ii) Athletes that are good in 100m tend to be also good in long jump.
(iii) The first principal component has the highest loadings for 100m and long jump.
(iv) 110m hurdle has a very small loading for PC2.

Solution FALSE - TRUE - TRUE - TRUE

Explanation: (i) No, because the loading of PC2 is negative for 1500m, thus a negative value for this loading means that the athlete will run long times! (ii) Yes, because 100m and long jump lie on the same axis for PC1 and PC2 - just in opposite directions, because low times in 100m corresponds to high widths in long jump. (iii) Yes, given that we look at the absolute values of the loadings. (iv) 110m hurdle is almost parallel to PC1.

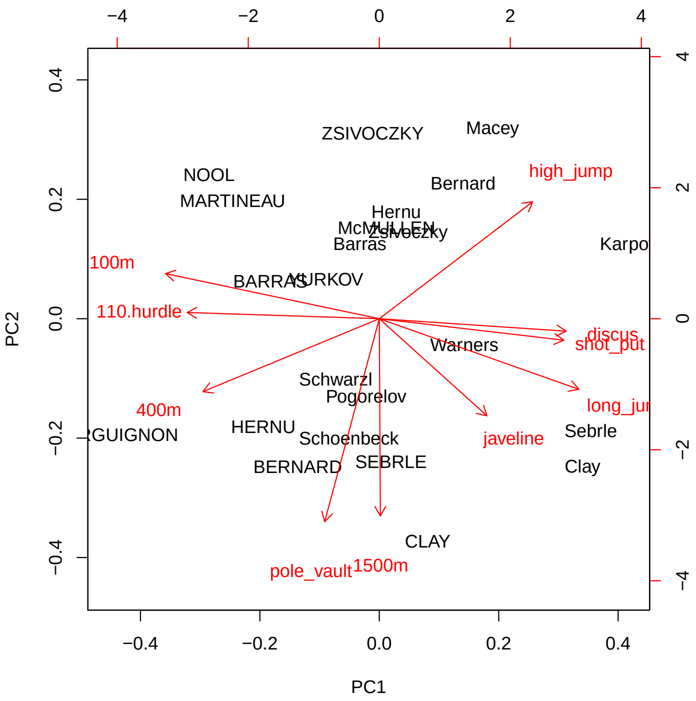


Figure 2: Figure for question 5g).