Compulsory Exercise 3

TMA4268 Statistical Learning V2020

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

21 mai, 2021

Last changes: 13.04.2020

Problem 1 (9P)

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

```
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.train)
```

```
## 'data.frame':    388 obs. of  18 variables:
##  $ Private    : num  1 2 1 2 2 1 2 2 2 2 ...
##  $ Apps       : num  1401 344 4216 427 2929 ...
##  $ Accept     : num  1239 264 2290 385 1834 ...
##  $ Enroll     : num  605 97 736 143 622 ...
##  $ Top10perc  : num  10 11 20 18 20 10 27 50 62 13 ...
##  $ Top25perc  : num  34 42 52 38 56 35 50 77 93 33 ...
##  $ F.Undergrad: num  3716 500 4296 581 2738 ...
##  $ P.Undergrad: num  675 331 1027 533 1662 ...
##  $ Outstate   : num  7100 12600 5130 12700 12600 ...
##  $ Room.Board : num  4380 5520 4690 5800 5610 ...
##  $ Books      : num  540 630 600 450 450 537 450 525 500 570 ...
##  $ Personal   : num  2948 2250 1450 700 3160 ...
##  $ PhD        : num  63 77 73 81 90 77 77 76 94 66 ...
##  $ Terminal   : num  88 80 75 85 90 84 98 92 96 83 ...
##  $ S.F.Ratio  : num  19.4 10.4 17.9 10.3 15.1 21 21.5 10.1 9.6 16 ...
##  $ perc.alumni: num  0 7 18 37 9 16 21 57 20 14 ...
##  $ Expend     : num  5389 9773 5125 11758 9084 ...
##  $ Grad.Rate  : num  36 43 56 84 84 54 64 77 93 66 ...
```

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!).

```
# Feature-wise normalization added to the predictors (not the response).
train.target <- college.train$Outstate
college.train <- subset(college.train, select = -c(Outstate))
test.target <- college.test$Outstate
college.test <- subset(college.test, select = -c(Outstate))

mean <- apply(college.train, 2, mean)
std <- apply(college.train, 2, sd)
college.train <- scale(college.train, center = mean, scale = std)
college.test <- scale(college.test, center = mean, scale = std)
```

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?

The equation which describes a network that predicts `Outstate` with 2 hidden layers and `relu` activation function with 64 units each, is

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

Since `Outstate` is a continuous variable, I would use a linear activation function for the output layer (for regression).

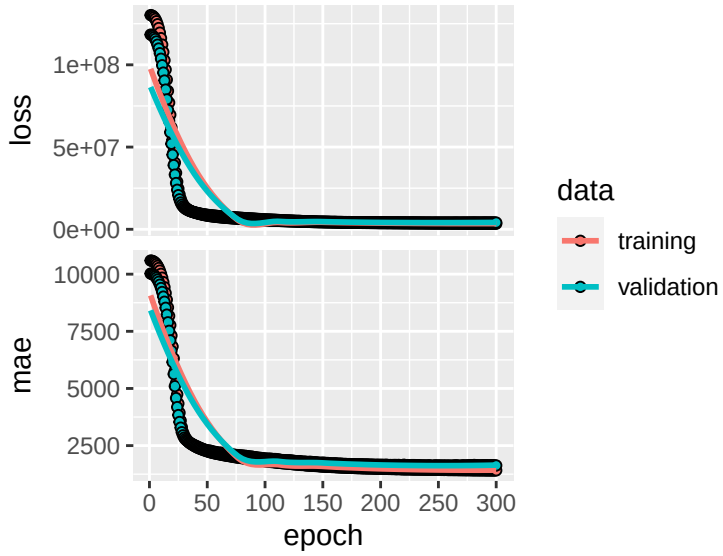
c) (3P)

(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).

```
set.seed(123)
# Build the model
model <- keras_model_sequential() %>% layer_dense(units = 64, activation = "relu",
    input_shape = ncol(college.train)) %>% layer_dense(units = 64, activation = "relu") %>%
    layer_dense(units = 1)

model %>% compile(optimizer = "rmsprop", loss = "mse", metrics = c("mae"))

# Train
history <- model %>% fit(college.train, train.target, epochs = 300, batch_size = 8,
    validation_split = 0.2, verbose = 0)
```

(ii) Plot the training and validation error as a function of the epochs (1P).

```
plot(history)
```



(iii) Report the MSE of the test set and compare it with methods that you used in Compulsory 2 (1P).

```
result <- model %>% evaluate(college.test, test.target, verbose = 0)
result[1]
```

```
##    loss
## 3733178
```

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.

```
set.seed(123)

# Regularization with weight decay with l2.  Build the model
model.kernel.reg <- keras_model_sequential() %>% layer_dense(units = 64, activation = "relu",
    input_shape = ncol(college.train), 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.kernel.reg %>% compile(optimizer = "rmsprop", loss = "mse", metrics = c("mae"))

# Train
history.kernel.reg <- model.kernel.reg %>% fit(college.train, train.target, epochs = 300,
    batch_size = 8, validation_split = 0.2, verbose = 0)

# Result for weight decay with l1.
result.kernel.reg <- model.kernel.reg %>% evaluate(college.test, test.target, verbose = 0)
result.kernel.reg[1]
```

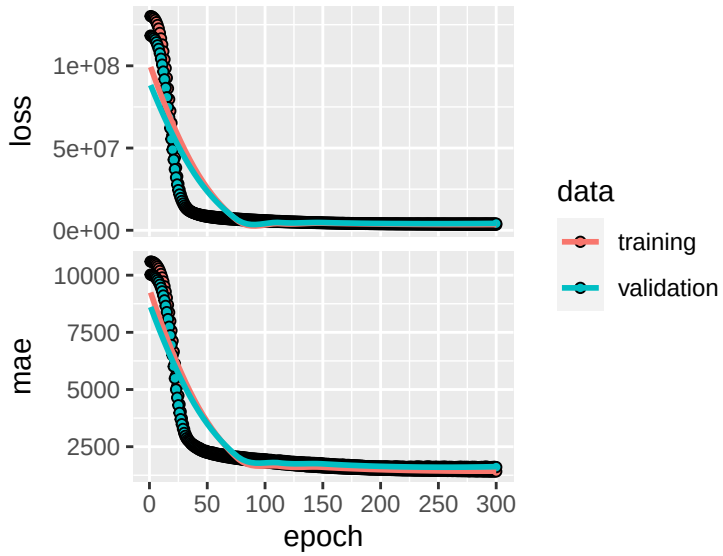
```
##    loss
## 3582584
```

```
# Regularization with dropout.  Build the model
model.drop.reg <- keras_model_sequential() %>% layer_dense(units = 64, activation = "relu",
```

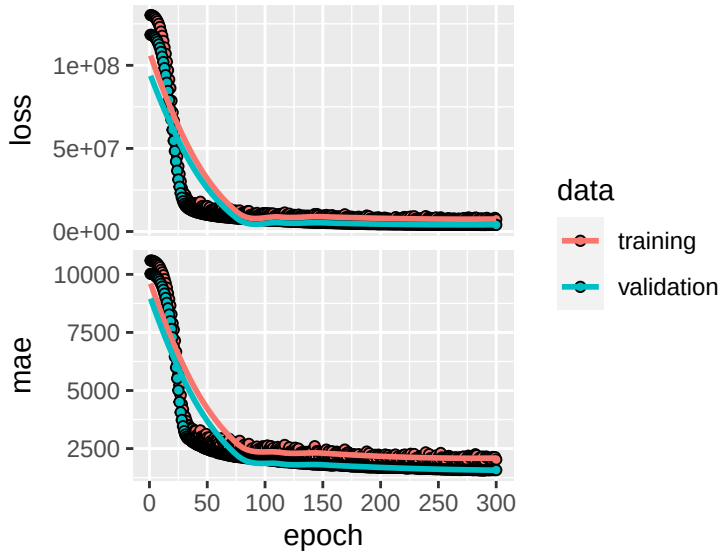
```
    input_shape = ncol(college.train)) %>% layer_dropout(rate = 0.4) %>% layer_dense(units = 64,
    activation = "relu") %>% layer_dropout(rate = 0.4) %>% layer_dense(units = 1)

model.drop.reg %>% compile(optimizer = "rmsprop", loss = "mse", metrics = c("mae"))

# Train
history.drop.reg <- model.drop.reg %>% fit(college.train, train.target, epochs = 300,
    batch_size = 8, validation_split = 0.2, verbose = 0)

# Result for dropout..
result.drop.reg <- model.drop.reg %>% evaluate(college.test, test.target, verbose = 0)
result.drop.reg[1]
```

```
##    loss
## 3489555
```

```
par(mfrow = c(1, 2))
plot(history.kernel.reg)
```




```
plot(history.drop.reg)
```




```
par(mfrow = c(1, 1))  # reset plotting grid.
```

The dropout seems to improve the model fit, while the regularization with weight decay with L2-norm does not seem to boost the performance compared to no use of regularization (from earlier).

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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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?

```
glm.fit <- glm(deceased ~ ., family = "binomial", data = d.corona)
summary(glm.fit)
```

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

```
## glm(formula = deceased ~ ., 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
```

```
anova(glm.fit, 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
```

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

Answer: FALSE

(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.

Answer: FALSE

(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.

Answer: FALSE Why is this false? I have asked the Q on Piazza. This is the case because the odds ratio is constantly equal to $\exp 10 \cdot \beta_{sex}$, while the odds changes with this amount (the odds ratio), when x is

changed by 10.

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

Answer: FALSE

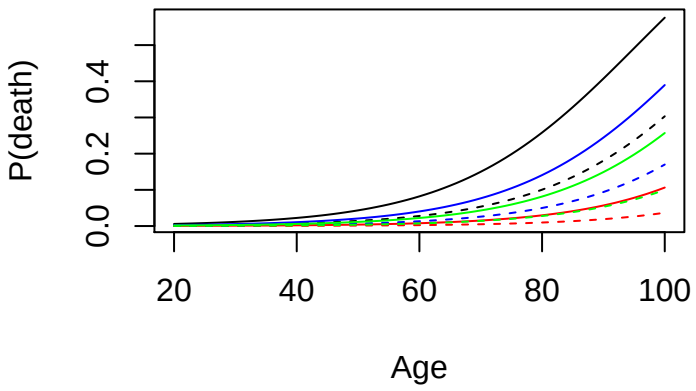
c) (1P)

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.

```
x.age <- seq(20, 100, 1)
y.male.france <- predict(glm.fit, newdata = expand.grid(sex = "male", age = x.age,
    country = "France"), type = "response")
y.female.france <- predict(glm.fit, newdata = expand.grid(sex = "female", age = x.age,
    country = "France"), type = "response")
y.male.japan <- predict(glm.fit, newdata = expand.grid(sex = "male", age = x.age,
    country = "japan"), type = "response")
y.female.japan <- predict(glm.fit, newdata = expand.grid(sex = "female", age = x.age,
    country = "japan"), type = "response")
y.male.indonesia <- predict(glm.fit, newdata = expand.grid(sex = "male", age = x.age,
    country = "indonesia"), type = "response")
y.female.indonesia <- predict(glm.fit, newdata = expand.grid(sex = "female", age = x.age,
    country = "indonesia"), type = "response")
y.male.korea <- predict(glm.fit, newdata = expand.grid(sex = "male", age = x.age,
    country = "Korea"), type = "response")
y.female.korea <- predict(glm.fit, newdata = expand.grid(sex = "female", age = x.age,
    country = "Korea"), type = "response")


plot(x.age, y.male.france, type = "l", ylab = "P(death)", xlab = "Age")
lines(x.age, y.female.france, lty = 2, col = "black")
lines(x.age, y.male.japan, col = "red")
lines(x.age, y.female.japan, lty = 2, col = "red")
lines(x.age, y.male.indonesia, col = "blue")
lines(x.age, y.female.indonesia, lty = 2, col = "blue")
lines(x.age, y.male.korea, col = "green")
lines(x.age, y.female.korea, lty = 2, col = "green")
```



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?

As shown from the full model in b), males do have a higher probability to die than females, since the odds for males is higher.

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

```
fit2 <- glm(deceased ~ . + age:sex, data = d.corona, family = "binomial")
summary(fit2)
```

```
##
## Call:
## glm(formula = deceased ~ . + age:sex, 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
```

We cannot conclude that this is the case, since the interaction between sex and age is not significant. Thus, we have no evidence that age is a greater risk for males than females, since the simple null hypothesis for the coefficient of the interaction term cannot be discarded.

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

```
fit3 <- glm(deceased ~ . + age:country, data = d.corona, family = "binomial")
summary(fit3)
```

```
##
## Call:
## glm(formula = deceased ~ . + 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
```

The same conclusion as in (ii): We cannot conclude that this is the case, since the interaction between Korea and age is not significant to any logical level.

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

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?

No, I think this sounds suspicious. I imagine that there has been run tests on more severe cases in France

compared to the ones run in the other cases, i.e. that people in France had more severe symptoms before testing.

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.

Answer: TRUE

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

Answer: TRUE

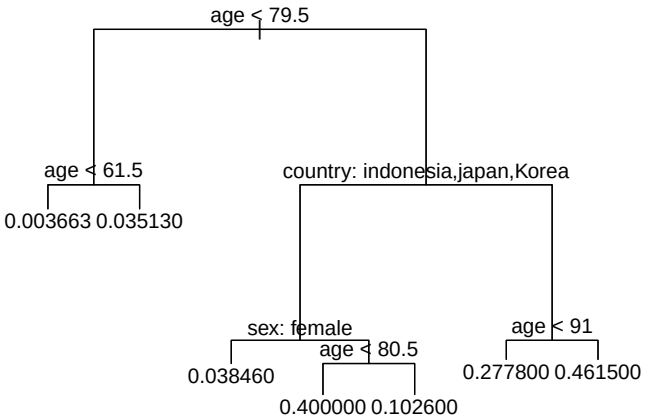
Consider the LDA code and output below:

(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.

Answer: TRUE (even though I get the null rate: 2.2900763 %).

(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.

Answer: FALSE




```
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
```

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):

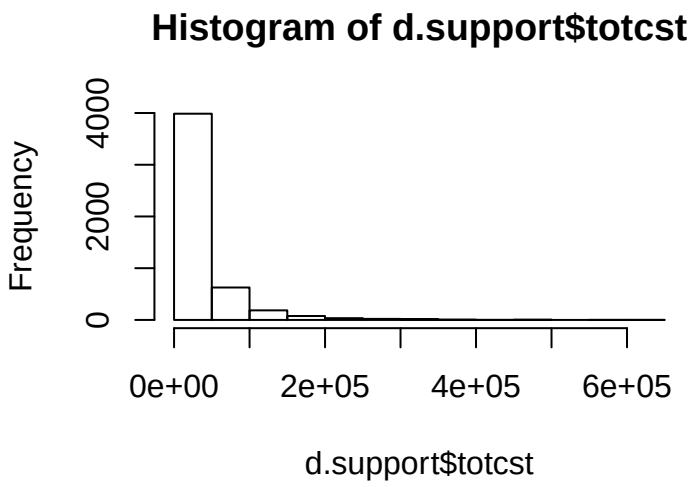
```
id <- "1heRtzi8vBoBGMaM2-ivBQI5Ki3HgJTmO"  # google file ID
d.support <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    id), header = T)
# We only look at complete cases
d.support <- d.support[complete.cases(d.support), ]
d.support <- d.support[d.support$totcst > 0, ]
```

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

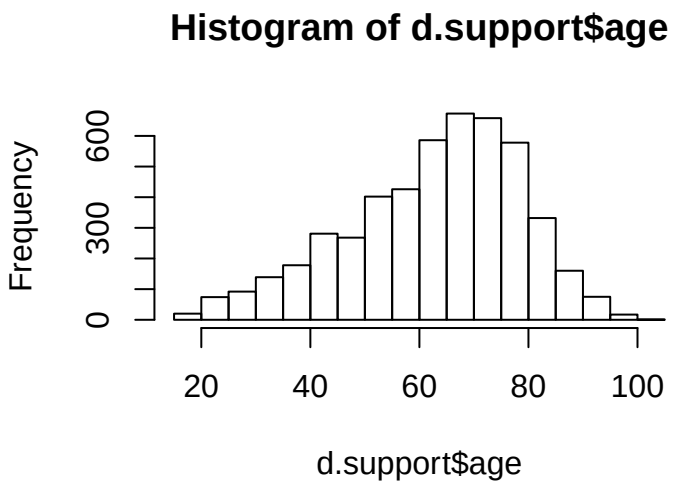
a) (1P)

Before we start analyzing 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.**

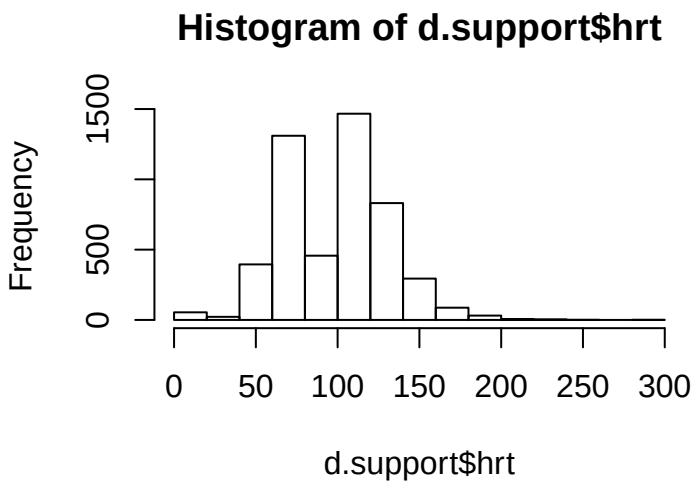
```
hist(d.support$totcst)
```



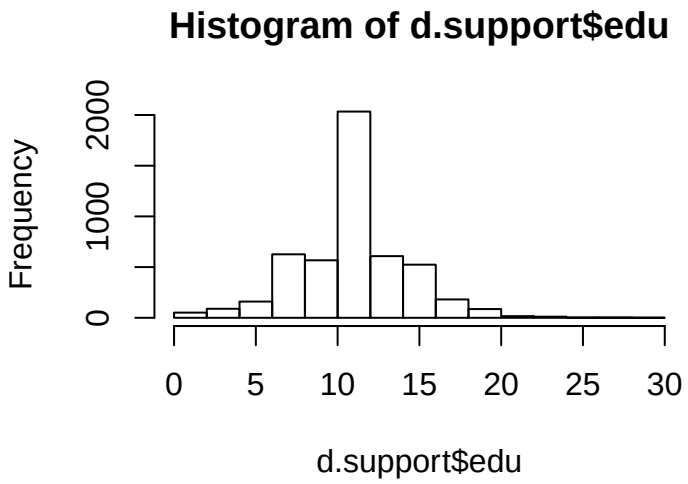

```
hist(d.support$age)
```



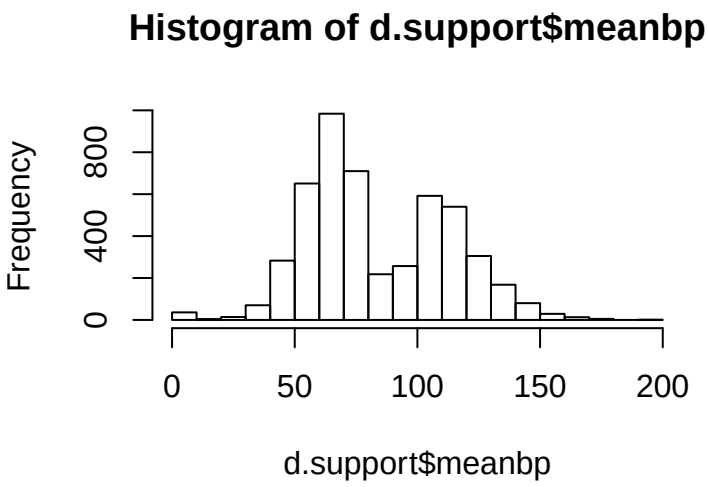

```
hist(d.support$hrt)
```



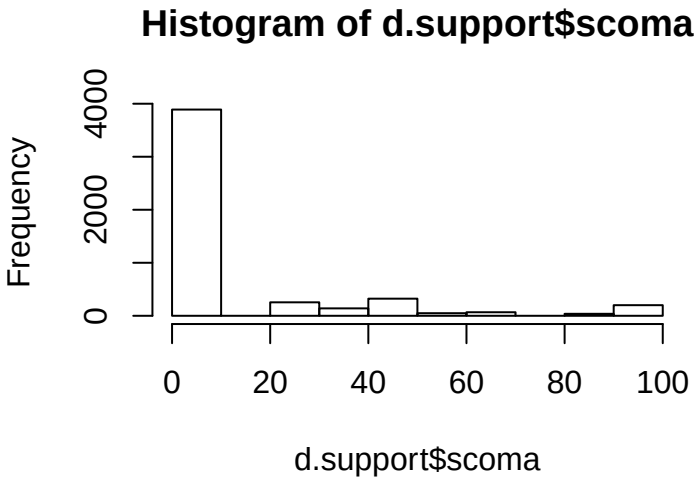

```
hist(d.support$edu)
```



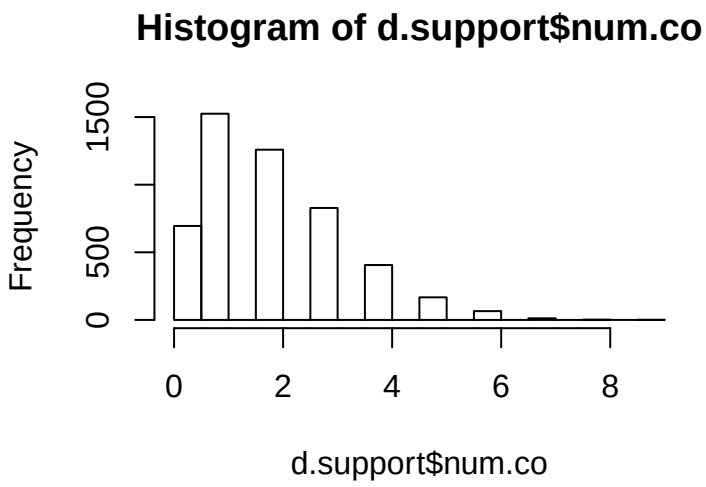

```
hist(d.support$meanbp)
```



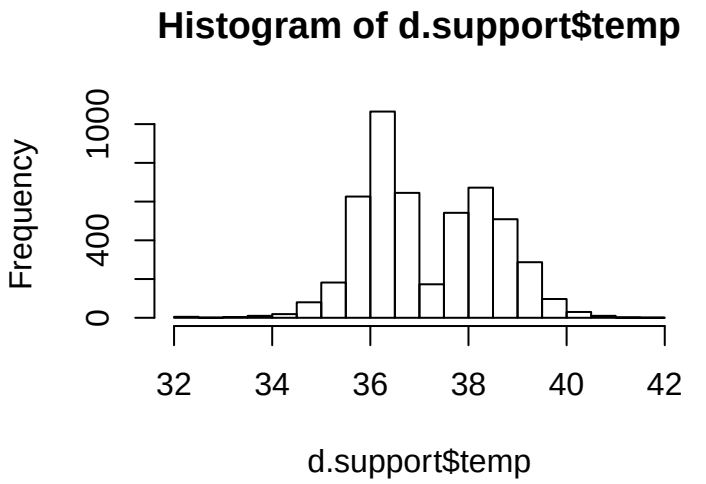

```r
hist(d.support$scoma)
```



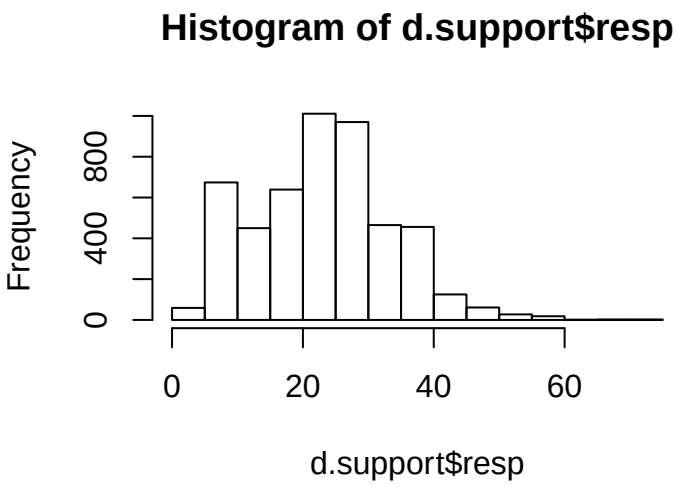

```r
hist(d.support$num.co)
```



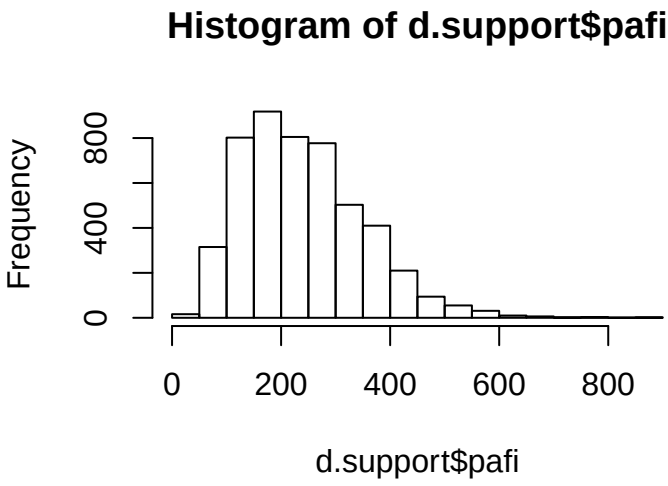

```
hist(d.support$temp)
```




```
hist(d.support$resp)
```




```
hist(d.support$pafi)
```



A log transform will be used since the distribution of totcst is skewed.

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`.

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

```
##
## Call:
## lm(formula = log(totcst) ~ age + temp + edu + resp + num.co +
##     dzgroup, data = d.support)
##
## Residuals:
```

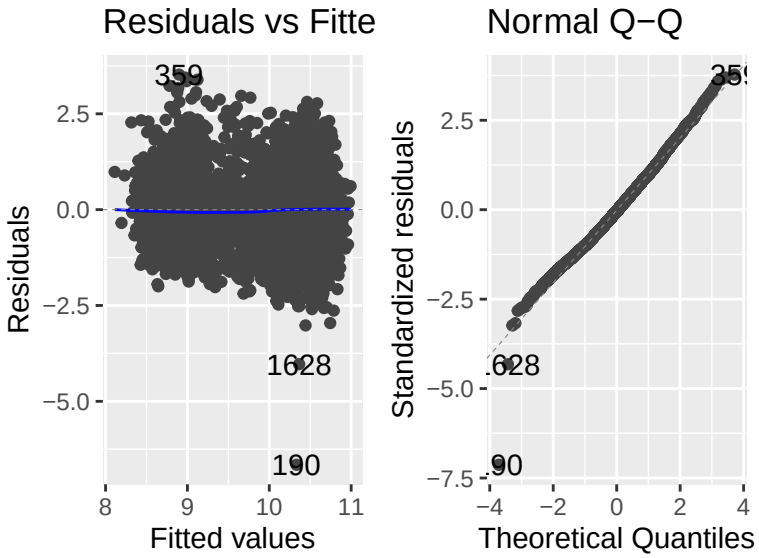
```
##     Min      1Q  Median      3Q     Max
## -6.6554 -0.6524 -0.0437  0.6203  3.5226
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             8.0823597  0.4014491  20.133  < 2e-16 ***
## age                    -0.0069950  0.0008742  -8.001 1.52e-15 ***
## 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
```

(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)

Answer: When a patient's age increases by 10 years, the total costs are expected to be reduced with $\exp(10 \cdot -0.0069950) \approx 0.932$

(ii) Do a residual analysis using the Tukey-Anscombe plot and the QQ-diagram. Are the assumptions fulfilled? (1P)

```
library(ggfortify)
autoplot(linear.fit, which = c(1, 2))
```



The assumptions of the linear model are fulfilled, when based on the two diagrams above, since the residual plot shows no clear pattern, while the QQ-diagram seems to behave nicely.

(iii) Does the effect of age depend on the disease group? Do a formal test and report the p-value. (1P)

```
dzgroup.age <- lm(log(totcst) ~ temp + edu + resp + num.co + dzgroup * age, data = d.support)
anova(dzgroup.age)
```

```
## Analysis of Variance Table
##
## Response: log(totcst)
##               Df Sum Sq Mean Sq  F value    Pr(>F)
## temp           1  238.6  238.59 274.8470 < 2.2e-16 ***
## edu            1  105.2  105.17 121.1507 < 2.2e-16 ***
## resp           1    4.0    3.98   4.5799 0.0323984 *
## num.co         1  321.4  321.45 370.2935 < 2.2e-16 ***
## dzgroup        7 1937.3  276.76 318.8136 < 2.2e-16 ***
## age            1   55.8   55.81  64.2943 1.327e-15 ***
## dzgroup:age    7   24.5    3.51   4.0387 0.0002019 ***
## Residuals   4940 4288.3    0.87
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

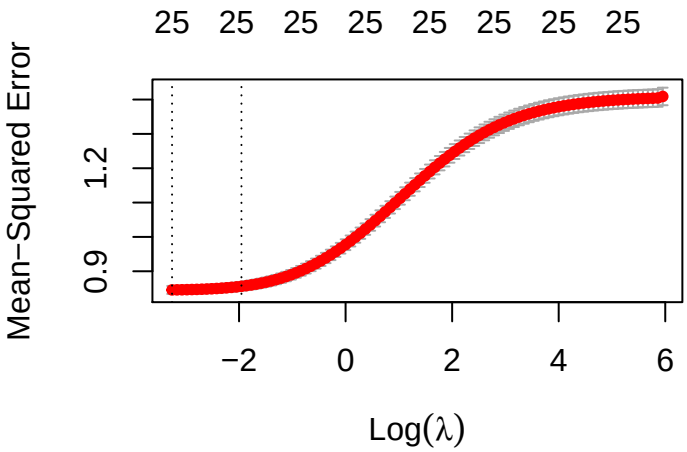
Yes, the p-value of the interaction term between dzgroup and age is 0.0002019, which is significant to a reasonable level. Hence, the effect of age depends on the disease group.

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).

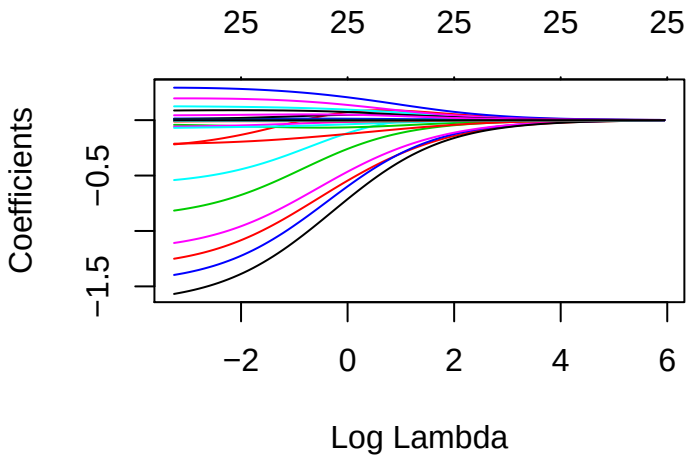
```
library(glmnet)
set.seed(12345)
train.ind = sample(1:nrow(d.support), 0.8 * nrow(d.support))
d.support.train = d.support[train.ind, ]
d.support.test = d.support[-train.ind, ]

x.train <- model.matrix(log(totcst) ~ ., data = d.support.train)[, -1]
y.train <- log(d.support.train$totcst)
x.test <- model.matrix(log(totcst) ~ ., data = d.support.test)[, -1]
y.test <- log(d.support.test$totcst)

set.seed(4268)
cv.ridge <- cv.glmnet(x.train, y.train, alpha = 0)
plot(cv.ridge)
```




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




```
ridge.pred <- predict(ridge, s = lambda.ridge, newx = x.test)
# MSE
mean((ridge.pred - y.test)^2)
```

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

The MSE is reported above.

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:

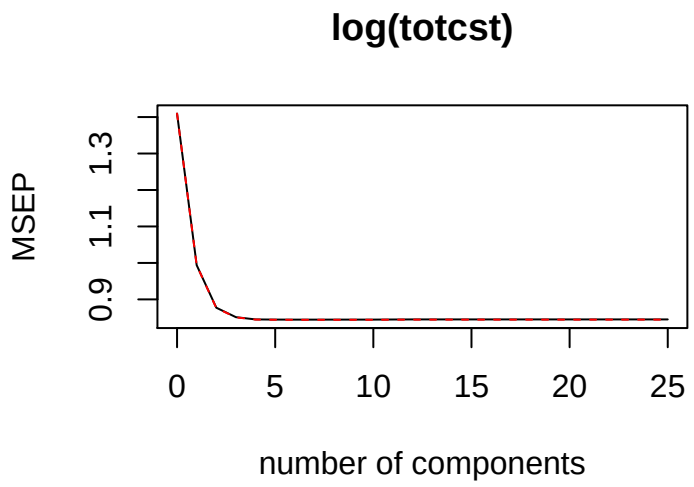
(i) Run a PLS regression (don't forget to scale the variables, `scale=TRUE`) (1P).
(ii) Choose an optimal number of principal components (PCs) using cross-validation (1P).
(iii) Report the MSE of the test set when using the respective set of PCs and compare to the result from ridge regression. Conclusion? (1P)

```
library(pls)
set.seed(234)
# PLS regression.
pls.fit <- plsr(log(totcst) ~ ., data = d.support.train, scale = TRUE, validation = "CV")
summary(pls.fit)
```

```
## 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  20 comps
## CV       0.9193    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    0.9190
##        21 comps  22 comps  23 comps  24 comps  25 comps
## CV       0.9193    0.9193    0.9193    0.9193    0.9193
## adjCV    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  14 comps
## X              38.94    43.78     47.74     50.34     52.94     55.23     57.47
## log(totcst)    40.75    40.75     40.75     40.75     40.75     40.75     40.75
##              15 comps  16 comps  17 comps  18 comps  19 comps  20 comps
## X               61.74     65.10     68.77     72.02     76.31     80.17
## log(totcst)     40.75     40.75     40.75     40.75     40.75     40.75
##              21 comps  22 comps  23 comps  24 comps  25 comps
## X               83.53     87.97     92.00     96.01    100.00
## log(totcst)     40.75     40.75     40.75     40.75     40.75
```

```r
validationplot(pls.fit, val.type = "MSEP")
```




```r
# Optimal number of PCs are 4 (Ockham's razor/when choosing the simplest model
# which is almost optimal)
pls.pred <- predict(pls.fit, d.support.test, ncomp = 4)
mean((pls.pred - log(d.support.test$totcst))^2)
```

```
## [1] 0.8638231
```

The MSE is slightly smaller in this case, compared to the result when using Ridge regression. Thus, PLS is a better choice than Ridge regression, even though the results do not differ by a lot, since prediction is the sole aim in this case.

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).

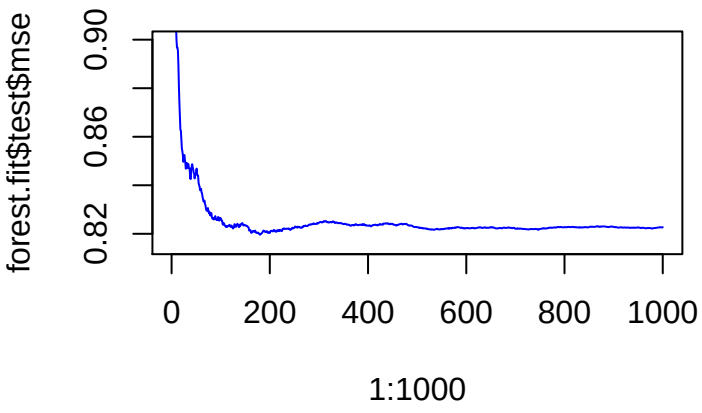
```
# GAM.
library(gam)
gam1 <- gam(log(totcst) ~ dzgroup + ns(age, 4) + num.co + race + income + s(edu,
    4) + poly(temp, 3) + ns(pafi, 5) + poly(scoma, 3) + s(meanbp) + ns(hrt, 3) +
    bs(resp, 5), data = d.support.train)
# plot(gam1)
pred.gam <- predict(gam1, newdata = d.support.test)
mean((pred.gam - log(d.support.test$totcst))^2)
```

```
## [1] 0.8363577
```

(ii) One model/method based on regression trees (2P).

```
# Random Forest.
library(randomForest)
set.seed(4268)
m <- round(ncol(d.support.train)/3)  # regression.
trees <- seq(from = 100, to = 800, by = 25)
mses <- rep(0, length(trees))
j <- 1
# for (i in trees){ # Try different amounts of trees B, to see where the error
# stabilizes. Did not really stabilize.  forest.fit <- randomForest(log(totcst) ~
# ., data = d.support.train, mtry = m, ntree = i, importance = T) forest.pred <-
# predict(forest.fit, newdata = d.support.test) mse <- mean((forest.pred -
# log(d.support.test$totcst))^2) mses[j] <- mse j <- j+1 } # very slow loop.
```

```
# plot(trees, mses, type = 'l', xlab = 'Number of trees B', ylab = 'MSE test')
# since they did not really stabilize, choose B = 1000 (e.g.)
train.predictors <- d.support.train[, -7]  # remove totcst
y.train <- log(d.support.train[, 7])  # only totcst
test.predictors <- d.support.test[, -7]  # remove totcst
y.test <- log(d.support.test[, 7])  # only totcst

# Way better method compared to the foor-loop above! (a lot more effective)
forest.fit <- randomForest(train.predictors, y = y.train, xtest = test.predictors,
    ytest = y.test, mtry = m, ntree = 1000, importance = T)
plot(1:1000, forest.fit$test$mse, col = "blue", type = "l", ylim = c(0.815, 0.9))
```




```
# Choose B = 1000.
forest.fit$test$mse[1000]
```

```
## [1] 0.8227119
```

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

The first choice was a GAM with several different non-linear predictors. These were chosen somewhat randomly in order to reduce the test MSE.

The second choice was a random forest, since this can be made better that bagging (less correlated trees) in many cases, and is simpler than boosting, which could also have been used.

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.

The basis functions of the model are (without the intercept)

$$X, X^2, X^3, (X-1)^3_+ \text{ and } (X-2)^3_+,$$

where

$$(X-q)^3_+ = \begin{cases} (X-q_j)^3 & , X > q_j \\ 0 & , \text{otherwise.} \end{cases}$$

The design matrix of the model is

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

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

The suitable methods for inference are (i), (ii) and (iii). SVMs are not suited for inference, since they are hard to interpret.

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).

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

```
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 -1.184477e-01 0.775427115
## t2*  1.13724644  1.608398e-02 0.355009899
## t3*  0.06801169  7.020207e-04 0.008216775
## t4* -0.75425940 -1.964887e+00 5.212417293
## t5* -2.43410057 -7.132773e-01 3.239046517
## t6* -1.36679680 -5.540903e-05 0.405064294
```

```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.

Answer: FALSE

(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.

Answer: TRUE. This is the case because the bootstrap is "always right", since it does not rely on any assumptions. Here, the data points might be dependent, which means that the SE is underestimated in the glm-function and the assumption of independent observation pairs in the logistic regression is broken.

(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.

Answer: TRUE. This is a consequence of the last point. Since the SE is underestimated, this means that the p-values are too small (since the T-values are too large).

(iv) The bootstrap relies on random sampling the same data without replacement.

Answer: FALSE

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

The following are techniques for regularization: (i), (ii) and (iv). Forward and backward selection are not techniques for regularization, since the estimated coefficient are not shrunk (only chosen). Why is Stochastic Gradient Descent a regularization technique? Look into this!

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.

Answer: FALSE

(ii) PLS involves the first principal components that are most correlated with the response.

Answer: TRUE

(iii) The idea in PLS is that we choose the principal components that explain most variation among all covariates.

Answer: FALSE

(iv) The idea in PCR is that we choose the principal components that explain most variation among all covariates.

Answer: 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.

Answer: The single correct statement is (iv).

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.

Answer: The single correct statement is (ii).

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}

Answer: The single correct statement is (i). Kom hit!

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.

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 pole_vault
## SEBRLE  11.04      7.58    14.83      2.07 49.81      14.69  43.75       5.02
## BERNARD 11.02      7.23    14.25      1.92 48.93      14.99  40.87       5.32
## YURKOV  11.34      7.09    15.19      2.10 50.42      15.31  46.26       4.72
##         javeline 1500m
## SEBRLE     63.19 291.7
## BERNARD    62.77 280.1
## YURKOV     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.

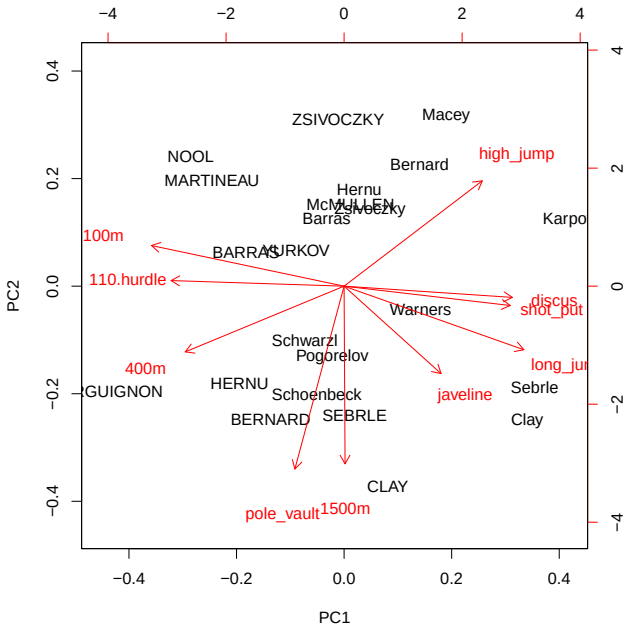


Figure 1: Figure for question 5g).