# Compulsory exercise 3

TMA4268 Statistical Learning V2019

Sara Elise Wøllo

03.05.2020

## Problem 1

### a)

```
set.seed(123)
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, ]

college.train.pr = college.train
college.test.pr = college.test
college.train.pr$Outstate = NULL  #Removing the Outstate column, it will be used in y_train
college.test.pr$Outstate = NULL  #Removing the Outstate column, it will be used in y_test
# normalize
mean <- apply(college.train.pr, 2, mean)
std <- apply(college.train.pr, 2, sd)
college.train.data <- scale(college.train.pr, center = mean, scale = std)
college.test.data <- scale(college.test.pr, center = mean, scale = std)
# Divide into redictors and response
x_train = college.train.data
x_test = college.test.data
y_train = college.train$Outstate
y_test = college.test$Outstate
```

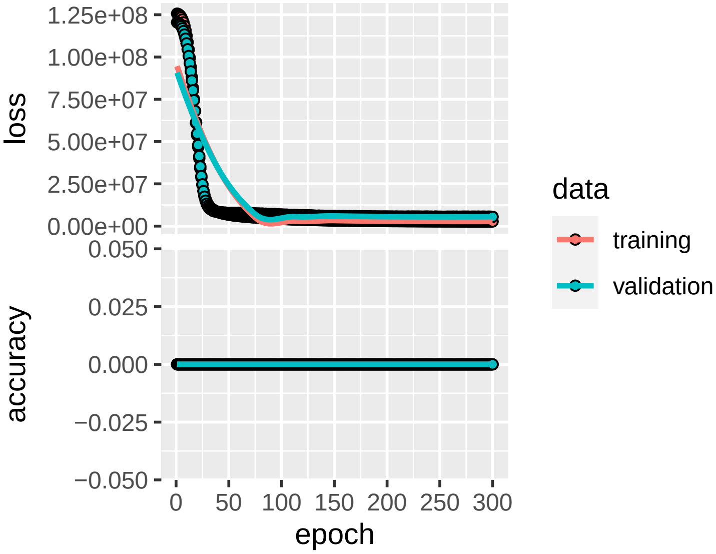
### b)

$$\hat{y_1} = \beta_{01} + \sum_{m=1}^{64} \beta_{m1} max\Big(\gamma_{0m} \sum_{l=1}^{64} \gamma_{lm} max\Big(\alpha_{0l} \sum_{j=1}^{17} \alpha_{jl} x_j, 0\Big), 0\Big)$$

Using linear activation on the output layer.

### c)

```
set.seed(123)

# Making the model
model <- keras_model_sequential()
model %>% layer_dense(units = 64, activation = "relu", input_shape = c(17)) %>% layer_dense(units = 64,
    activation = "relu") %>% layer_dense(units = 1, activation = "linear")
# summary(model)

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

history = model %>% fit(x_train, y_train, epochs = 300, batch_size = 8, validation_split = 0.2)
plot(history)
```



```
mse_nn = model %>% evaluate(x_test, y_test)
mse_nn[1]
```

```
## $loss
## [1] 5369839
```

We see that the MSE is 5369839, which is larger than the methods uses in compulsory 2, indicating a wrse fit.

### d)

```
set.seed(123)

# Making the model
model <- keras_model_sequential()
model %>% layer_dense(units = 64, activation = "relu", kernel_regularizer = regularizer_l1_l2(l1 = 0.01
    l2 = 0.05), input_shape = c(17)) %>% layer_dense(units = 64, activation = "relu",
    kernel_regularizer = regularizer_l1_l2(l1 = 0.01, l2 = 0.05)) %>% layer_dense(units = 1,
    activation = "linear")
# summary(model)

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

history = model %>% fit(x_train, y_train, epochs = 300, batch_size = 8, validation_split = 0.2)
# MSE
mse_nn_l1 = model %>% evaluate(x_test, y_test)
mse_nn_l1[1]
```

```
## $loss
## [1] 5255636
```

This improves the network slightly, to an MSE of 5255636, but it is not a large improvement.

# Problem 2

## a)

```
count(d.corona, country, deceased)
```

```
## # A tibble: 8 x 3
##   country   deceased     n
##   <fct>        <int> <int>
## 1 France           0   100
## 2 France           1    14
## 3 indonesia        0    67
## 4 indonesia        1     2
## 5 japan            0   291
## 6 japan            1     3
## 7 Korea            0  1507
## 8 Korea            1    26
```

```
count(d.corona, sex, deceased)
```

```
## # A tibble: 4 x 3
##   sex    deceased     n
##   <fct>     <int> <int>
## 1 female        0  1075
## 2 female        1    14
## 3 male          0   890
## 4 male          1    31
```

```
count(d.corona, country, sex, deceased)
```

```
## # A tibble: 15 x 4
##    country   sex    deceased     n
##    <fct>     <fct>     <int> <int>
##   1 France    female        0    55
##   2 France    female        1     5
##   3 France    male          0    45
##   4 France    male          1     9
##   5 indonesia female        0    29
##   6 indonesia female        1     1
##   7 indonesia male          0    38
##   8 indonesia male          1     1
##   9 japan     female        0   120
##  10 japan     male          0   171
##  11 japan     male          1     3
##  12 Korea     female        0   871
##  13 Korea     female        1     8
##  14 Korea     male          0   636
##  15 Korea     male          1    18
```

We see that there are 14 (5 female (f), 9 male (m)) deceased in France, 2 (1 (f), 1 (m)) in Indonesia, 3 (0 (f), 3 (m)) in Japan and 26 ( 8(f), 18 (m)) in Korea. There are 14 females deceased and 31 males deceased.

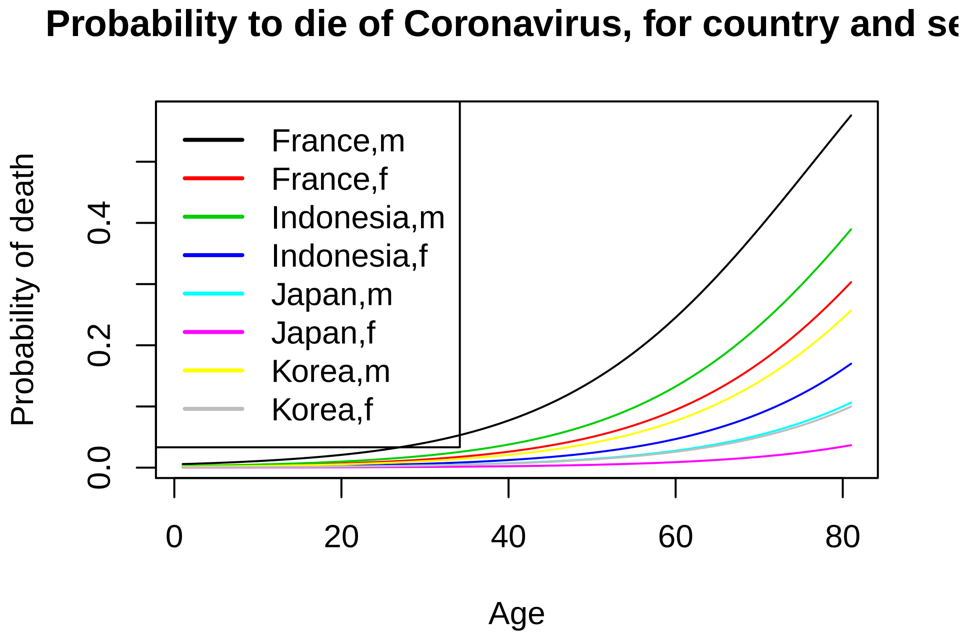
## b)

```
fit <- glm(deceased ~ sex + age + country, data = d.corona, family = "binomial")
# ggplot(data = d.corona, aes(age, deceased)) + geom_point() summary(fit)
```

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

## c)

```
plot(m_france, type = "l", lwd = 1, col = 1, xlab = "Age", ylab = "Probability of death")
lines(f_france, lwd = 1, col = 2)
lines(m_indo, lwd = 1, col = 3)
lines(f_indo, lwd = 1, col = 4)
lines(m_japan, lwd = 1, col = 5)
lines(f_japan, lwd = 1, col = 6)
lines(m_korea, lwd = 1, col = 7)
lines(f_korea, lwd = 1, col = 8)
title("Probability to die of Coronavirus, for country and sex")
legend(x = "topleft", legend = c("France,m", "France,f", "Indonesia,m", "Indonesia,f",
    "Japan,m", "Japan,f", "Korea,m", "Korea,f"), lwd = c(2, 2, 2, 2, 2, 2, 2, 2),
    col = c(1, 2, 3, 4, 5, 6, 7, 8), y.intersp = 1)
```



### d)

```
fit <- glm(deceased ~ sex + age, data = d.corona, family = "binomial")
fit_country <- glm(deceased ~ age + country, data = d.corona, family = "binomial")
fit_sex <- lm(deceased ~ sex, data = d.corona)
```

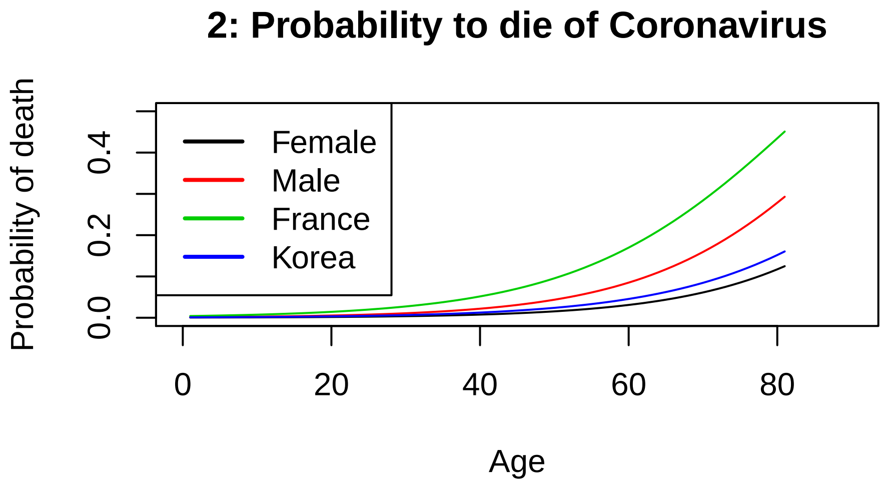
```
# Probability of men to die of Coronavirus
male_pred
```

```
##          1
## 0.03365907
```

```
# Probability of women to die of Coronavirus
female_pred
```

```
##          1
## 0.01285583
```

```
plot(f_pred, type = "l", lwd = 1, col = 1, xlab = "Age", ylab = "Probability of death",
    xlim = c(0, 90), ylim = c(0, 0.5))
lines(m_pred, lwd = 1, col = 2)
lines(france_pred, lwd = 1, col = 3)
lines(korea_pred, lwd = 1, col = 4)
title("2: Probability to die of Coronavirus")
legend(x = "topleft", legend = c("Female", "Male", "France", "Korea"), lwd = c(2,
    2, 2, 2), col = c(1, 2, 3, 4), y.intersp = 1)
```



i) True. Probability for men to die is 3.4 percent, whereas probability for women is 1.3 percent. You can also see from the plot "2: Probability to die of Coronavirus" that Males have higher probability of death than women at all ages.

ii) True. At a low age, the mortality rates are similar, but at age increases, the mortality rates increases faster for men than for women.

iii) True. The mortality rate for the Frence population is higher even at low ages, but the difference increases as age increaces.

### e)

Without knowing how the data was collected, this is not a result we can trust. We don't know how many were tested, and how sick people needed to be to be tested. If France only tested the people that were hospitalized, and the other countries tested more people with milder symptons, then it makes sense for France to have a higher mortality rate.

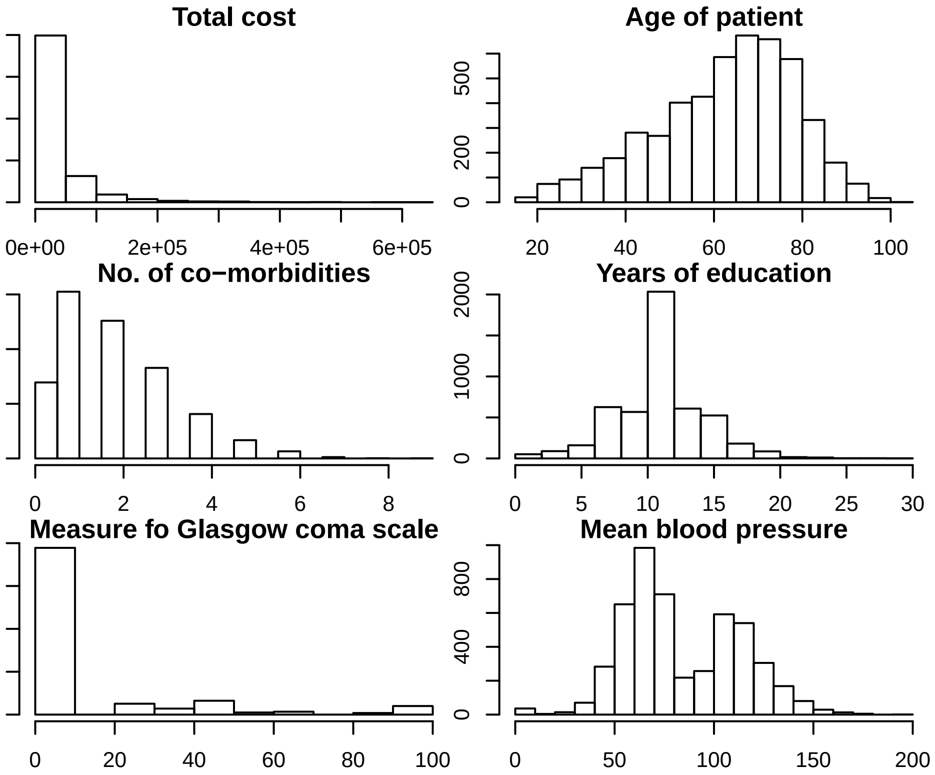
### f)

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

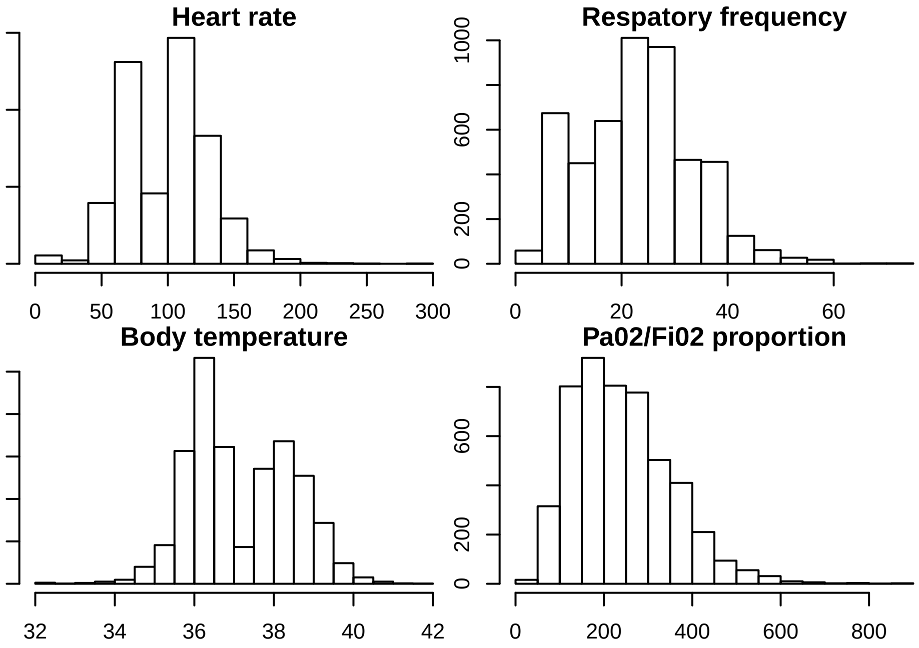
## Problem 3

### a)

```r
par(mfrow = c(3, 2), mar = c(2, 1, 1, 1))
hist(d.support$totcst, plot = TRUE, main = "Total cost")
hist(d.support$age, plot = TRUE, main = "Age of patient")
hist(d.support$num.co, plot = TRUE, main = "No. of co-morbidities")
hist(d.support$edu, plot = TRUE, main = "Years of education")
hist(d.support$scoma, plot = TRUE, main = "Measure fo Glasgow coma scale")
hist(d.support$meanbp, plot = TRUE, main = "Mean blood pressure")
```

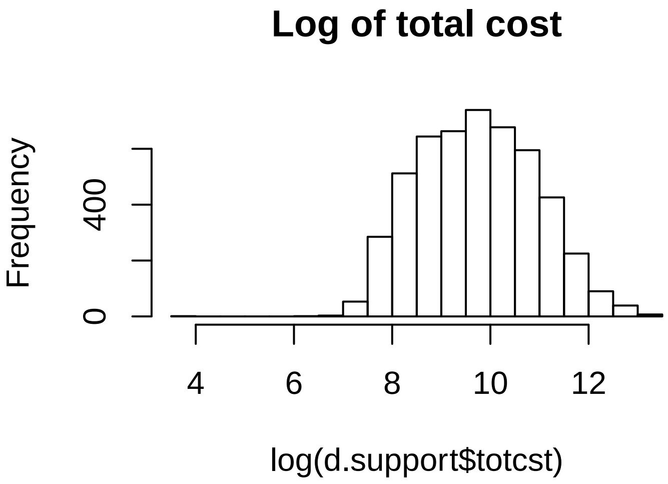


```r
par(mfrow = c(3, 2), mar = c(2, 1, 1, 1))
hist(d.support$hrt, plot = TRUE, main = "Heart rate")
hist(d.support$resp, plot = TRUE, main = "Respatory frequency")
hist(d.support$temp, plot = TRUE, main = "Body temperature")
hist(d.support$pafi, plot = TRUE, main = "Pa02/Fi02 proportion")
```



I suggest a logarithmic transformation to the variable totcst, histogram seen here:

```r
hist(log(d.support$totcst), plot = TRUE, main = "Log of total cost")
```



```r
# From now, we use the transformed version of totcst
d.support$totcst = log(d.support$totcst)
```

### b)

```r
fit = glm(totcst ~ age + temp + edu + resp + num.co + dzgroup, data = d.support)
# summary(fit)
```

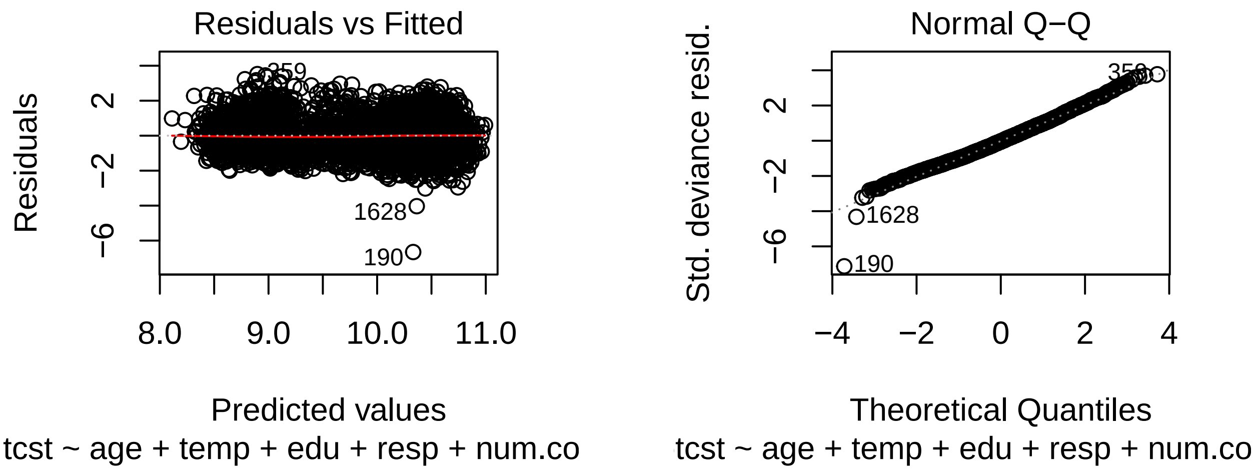
```r
new_grid = expand.grid(age = c(10, 20, 30, 40, 50, 60, 70, 80, 90), temp = 36, edu = 10,
    resp = 20, num.co = 2, dzgroup = "CHF")

cost = exp(predict.glm(fit, newdata = new_grid, type = "response"))
cost
```

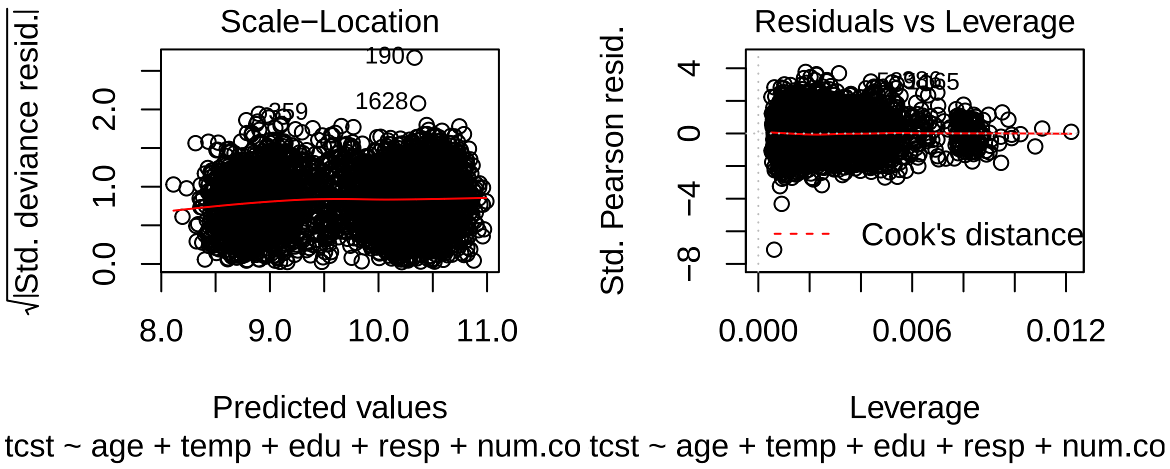
```
##        1        2        3        4        5        6        7        8 
## 9954.460 9281.939 8654.854 8070.134 7524.917 7016.536 6542.500 6100.491 
##        9 
## 5688.343 
```

i) When a patient's age increases by 10 years, the cost increase by factor 0.93244 (or equivalently, decrease by factor 1.07245).

ii)

```r
plot(fit)
```





We see from the Q-Q-diagram that the distibrution is normal, and not skewed. We see from the residuals vs. fitted-plot that there is no clear pattern, indicating that the assumptions in the model are fulfilled.

iii)

```
# Interaction term
fit = glm(totcst ~ age + temp + edu + resp + num.co + dzgroup + age * dzgroup, data = d.support)
summary(fit)$coefficients[, 4]
```

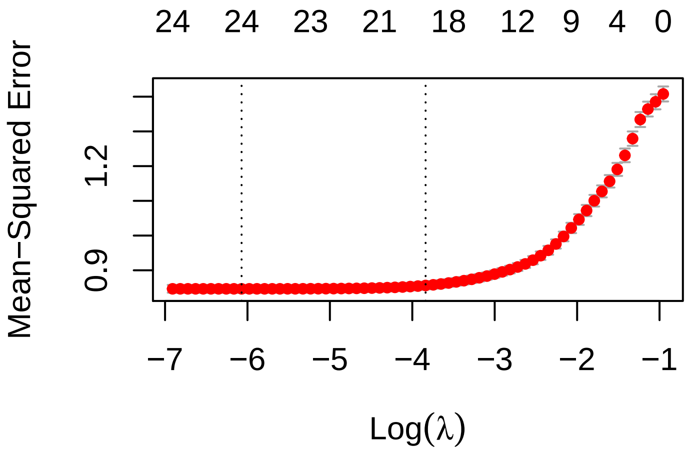
```
##             (Intercept)                     age                    temp
##            6.493813e-82            3.093651e-05            2.612091e-11
##                     edu                    resp                  num.co
##            3.824789e-10            3.419613e-02            1.011502e-04
##              dzgroupCHF       dzgroupCirrhosis     dzgroupColon Cancer
##            3.601977e-08            3.137701e-02            1.642433e-04
##             dzgroupComa             dzgroupCOPD      dzgroupLung Cancer
##            2.171373e-01            6.540501e-09            1.338537e-10
##       dzgroupMOSF w/Malig          age:dzgroupCHF    age:dzgroupCirrhosis
##            6.562770e-02            4.251551e-02            1.666083e-01
## age:dzgroupColon Cancer         age:dzgroupComa         age:dzgroupCOPD
##            3.253053e-01            5.887614e-04            4.206100e-01
##  age:dzgroupLung Cancer age:dzgroupMOSF w/Malig
##            4.555332e-01            1.053317e-03
```

Yes, we can see that the effect of age depends on the disease group. for Coma patients and MOSF-patients, the p-values suggest that the interaction is significant, whereas for other diseases, p-value suggests that there is not as big of an age effect.

### c)

```
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, ]
lambdas <- 10^seq(2, -3, by = -0.1)
x_train <- model.matrix(totcst ~ ., data = d.support.train)
y_train <- d.support.train$totcst
ridge_mod <- glmnet(x_train, y_train, family = "gaussian", alpha = 0)
cv.out <- cv.glmnet(x_train, y_train, aplha = 0)
plot(cv.out)
```



```
lambda_1se <- cv.out$lambda.1se
lambda_1se
```

```
## [1] 0.02152102
```

The largest value of lambda such that the eror is within 1 std.error of the smallest lambda is 0.02152102.

```
x_test <- model.matrix(totcst ~ ., data = d.support.test)
y_test <- d.support.test$totcst
ridge_pred <- predict(ridge_mod, s = lambda_1se, newx = x_test)
mse_ridge <- mean(as.numeric((ridge_pred - y_test)^2))
mse_ridge
```

```
## [1] 0.8636056
```

The test MSE of the ridge regression using lambda = 0.02152102 is 0.8636056.

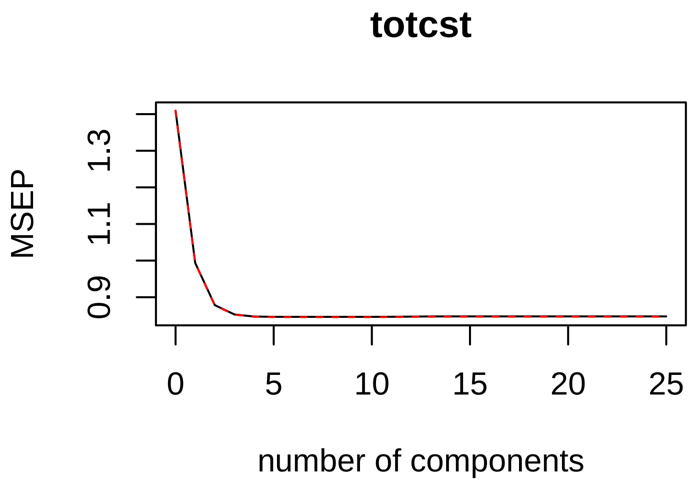
## d)

i)

```
set.seed(1)
pls_mod <- plsr(totcst ~ ., data = d.support.train, scale = TRUE, validation = "CV")
```

ii)

```
validationplot(pls_mod, val.type = "MSEP")
```



```
# MSEP(pls_mod)
selectNcomp(pls_mod, method = "onesigma")
```

```
## [1] 3
```

From the standard error of the CV residuals, we find that the best no. of componenets is 3.

iii)

```
pls_pred = predict(pls_mod, d.support.test, ncomp = 3)
mse_pls <- mean(as.numeric((pls_pred - d.support.test$totcst)^2))
mse_pls
```

```
## [1] 0.8664414
```

The MSE of the test set when using 3 PCs are 0.8664414.

```
mse_ridge
```

```
## [1] 0.8636056
```

```
mse_pls
```

```
## [1] 0.8664414
```

The MSE is similar, but the MSE from PLS is slightly higher. One is not significantly better than the other.

## e)

i)

```
fitgam = gam(totcst ~ bs(age, knots = c(40, 60, 80)) + poly(num.co, 3) + s(edu, df = 5) +
    income + race + s(meanbp, df = 5) + s(hrt, df = 5) + bs(resp, knots = c(20)) +
    bs(temp, knots = c(35, 37, 38)) + poly(pafi, 2) + bs(scoma, knots = c(10, 30)) +
    dzgroup, data = d.support.train)
```

```
# summary(fitgam) not added, as there was not enough space
gam_pred = predict(fitgam, d.support.test)
gam_sq_err <- as.numeric((gam_pred - d.support.test$totcst)^2)
mse_gam = mean(gam_sq_err)
mse_gam
```

```
## [1] 0.8330579
```

The choices of which transformation of the covariates was chosen, were done by plotting the different covariates and finding a suitable transfomation, depending on the spread of the data. Also, I spent some time trying out how the different transformations affected the different covariates. But, using this GAM, the MSE was 0.8330579.

ii)

```
# Bagging with random forest
set.seed(1)
oob.err = double(13)
mse_bag = double(13)
ntree = 350
for (mtry in 1:13) {
    fit = randomForest(totcst ~ ., data = d.support.train, mtry = mtry, ntree = ntree,
        importance = TRUE)
    oob.err[mtry] = fit$mse[ntree]
    pred = predict(fit, d.support.test)
    mse_bag[mtry] = with(d.support.test, mean((d.support.test$totcst - pred)^2))
}
min(mse_bag)
```

```
## [1] 0.8181038
```

Using bagging, we find that the MSE is 0.8181038.

This is the mest model fitted in this exercise. Bagging is suitable for regression tree problems like this is. I did this because the result from a standard regression tree was too poor.

# Problem 4

## a)

Basis functions:

$$b_1(x) = X,\ b_2(x) = X^2,\ b_3(x) = X^3,\ b_4(x) = (X-1)^3_+,\ b_5(x) = (X-2)^3_+.$$

Design matrix:

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

## b)

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

## c)

i) False ii) False iii) False iv) False

# Problem 5

## a)

i) True, ii) True, iii) False, iv) False

## b)

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

## c,d,e)

True: c: iv), d: ii), e: iv)

## f)

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

## g)

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