Compulsory Exercise 3

TMA4268 Statistical Learning

Helge Bergo

23 April, 2020

Problem 1

a)

Using the `College` data set, the training and test data was preprocessed, by separating the response and predictors into an x-matrix and y-vector for each set, and then scaling the predictors.

```
y.train = college.train$Outstate
y.test = college.test$Outstate

x.train <- subset(college.train, select = -c(Outstate))
x.test <- subset(college.test, select = -c(Outstate))

mean <- apply(x.train, 2, mean)
std <- apply(x.train, 2, sd)

x.train <- as.array(scale(x.train, center = mean, scale = std))
x.test <- as.array(scale(x.test, center = mean, scale = std))
```

b)

The equation for the network to predict `Outstate`, using an input layer with the 17 predictors and a `relu` activation function for the hidden layers is:

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

The activation function chosen for the output layer was the `linear` function, since this is a regression problem.

c)

(i)

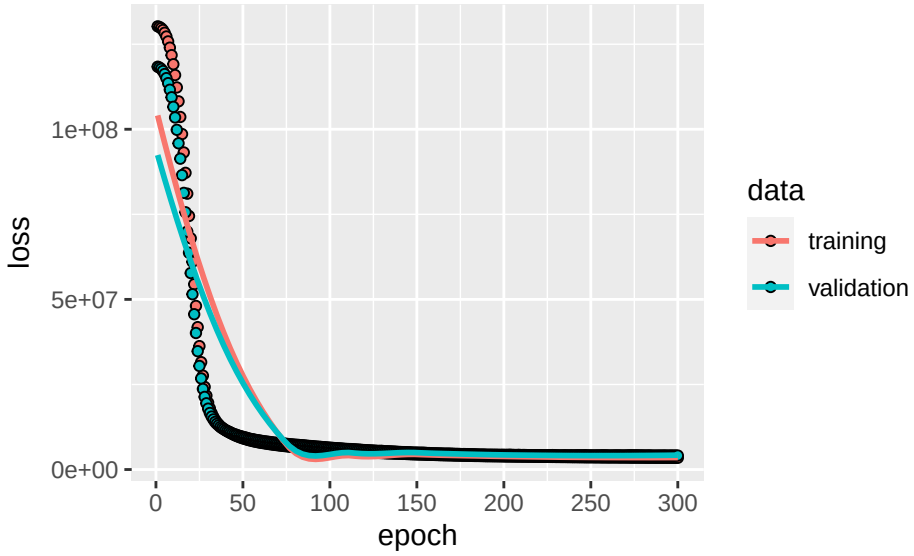
The network was trained using the `keras` library, using the chosen `linear` function for the output layer, and `mse` as the loss function.

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

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

history <- model %>%
  fit(x.train, y.train, epochs = 300, batch_size = 8, validation_split = 0.2)
```

(ii)

After training for 300 epochs, with 20% of the training data as the validation set, the results are plotted below.

```
plot(history)
```



As can be seen, both the training and validation error falls very quickly the first 30 epochs, and then continue to decrease slowly throughout the training.

(iii)

```
score <- model %>%
  evaluate(x.test, y.test)
```

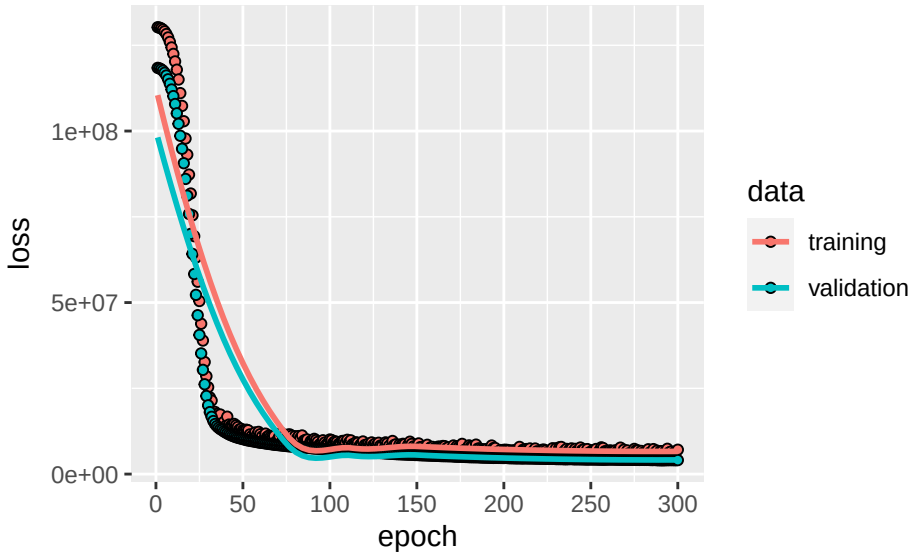
The final MSE after training the model for 300 epochs was 3.6×10^6. Compared to the MSE of the methods from Compulsory 2, this is a relatively good MSE score, and compares to both lasso and forward selection. It is better than polynomial regression and smoothing splines, but both bagging and random forest beat it, scoring 3.3×10^6 and 2.6×10^6 respectively.

d)

Both dropout and weight decay was tried out for improving the performance of the network.

```
model_reg <- keras_model_sequential() %>%
  layer_dense(units = 64, activation = "relu", input_shape = c(17),
              kernel_regularizer = regularizer_l2(l = 0.001)) %>%
  layer_dropout(0.3) %>%
  layer_dense(units = 64, activation = "relu") %>%
  layer_dropout(0.3) %>%
  layer_dense(units = 1, activation = "linear")

model_reg %>%
  compile(optimizer = "rmsprop", loss = "mse")

history_reg <- model_reg %>%
  fit(x.train, y.train, epochs = 300, batch_size = 8, validation_split = 0.2)
```

```
plot(history_reg)
```




```
score_reg <- model_reg %>% evaluate(x.test, y.test)
```

After implementing 30% dropout for the two hidden layers, and L2 regularization for the first hidden layer, the final MSE after training was 3.5×10^6, so lower than the unimproved network, but still not better than random forest, for example.

Problem 2

a) Inspecting your data

Table 1: Number of deceased per country.

country	n
France	14
indonesia	2
japan	3
Korea	26

Table 2: Number of deceased per sex.

sex	n
female	14
male	31

Table 3: Number of deceased per country, separated by gender.

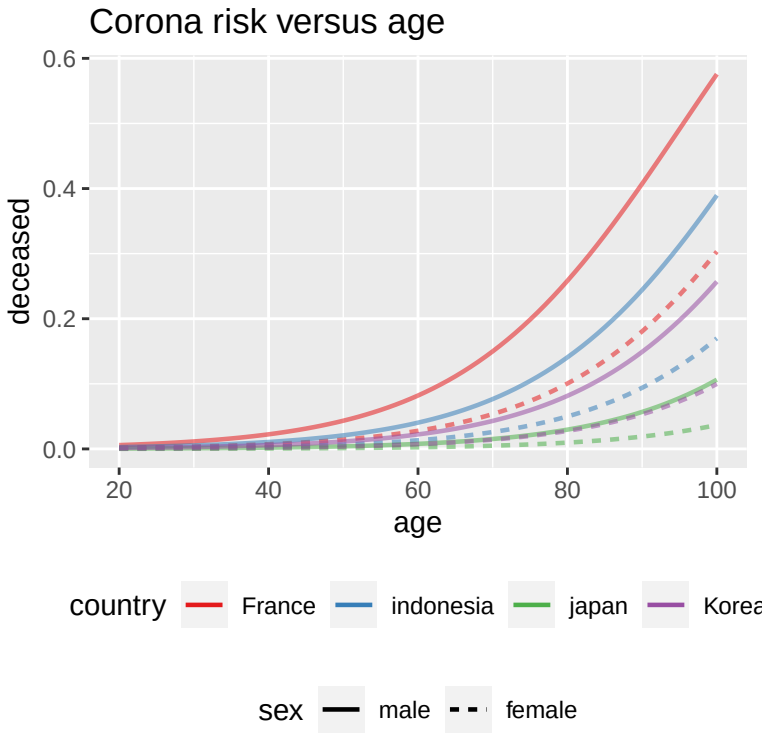
country	male	female
France	9	5
japan	3	0
indonesia	1	1
Korea	18	8

b) Multiple choice

FALSE, FALSE, TRUE, TRUE

c)

```
newdat = expand.grid(age = seq(20,100,1), sex = c('male','female'), country =
                       unique(d.corona$country))
newdat$pred = predict(glm_model, newdata = newdat, type = 'response')

d.corona %>%
  ggplot(aes(x = age, y = deceased, colour = country, lty = sex, alpha = 0.8)) +
  geom_line(data = newdat, aes(y = pred), size = 0.8) +
  labs(title = "Corona risk versus age")  +
  theme(legend.position = "bottom", legend.box = 'vertical') +
  scale_alpha(guide = 'none') +
  scale_color_brewer(palette = "Set1")
```



d)

i)

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

```
d.corona %>%
  glm(deceased ~ sex, data = ., family = 'binomial') %>%
  summary
```

```
##
## Call:
## glm(formula = deceased ~ sex, family = "binomial", data = .)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.2617  -0.2617  -0.1609  -0.1609   2.9509
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.3410     0.2690 -16.138  < 2e-16 ***
## sexmale       0.9838     0.3252   3.025  0.00248 **
## ---
## 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: 420.95  on 2008  degrees of freedom
## AIC: 424.95
```

```
##
## Number of Fisher Scoring iterations: 7
```

ii)

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

```
d.corona %>%
  glm(deceased ~ sex * age, data = ., family = 'binomial') %>%
  summary
```

```
##
## Call:
## glm(formula = deceased ~ sex * age, family = "binomial", data = .)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7976  -0.2015  -0.1079  -0.0568   3.3656
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.280111   1.365429  -6.796 1.07e-11 ***
## sexmale      1.386686   1.648811   0.841    0.400
## age          0.073877   0.016745   4.412 1.02e-05 ***
## sexmale:age -0.004067   0.020485  -0.199    0.843
## ---
## 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: 340.34  on 2006  degrees of freedom
## AIC: 348.34
##
## Number of Fisher Scoring iterations: 8
```

no write more here ### iii) (iii) Is age a greater risk factor for the French population than for the Korean population?

```
d.corona %>%
  glm(deceased ~ country * age, data = ., family = 'binomial') %>%
  summary
```

```
##
## Call:
## glm(formula = deceased ~ country * age, family = "binomial",
##     data = .)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.27939  -0.18381  -0.11772  -0.05523   3.08865
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -9.22100    2.35914  -3.909 9.28e-05 ***
```

```
## countryindonesia      5.29256    3.14368   1.684 0.092268 .
## countryjapan          2.91048    3.21279   0.906 0.364987
## countryKorea          0.73700    2.53247   0.291 0.771036
## age                   0.09553    0.02804   3.406 0.000658 ***
## countryindonesia:age -0.08735    0.04659  -1.875 0.060800 .
## countryjapan:age     -0.06736    0.04212  -1.599 0.109730
## countryKorea:age     -0.02660    0.03035  -0.876 0.380777
## ---
## 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: 328.01  on 2002  degrees of freedom
## AIC: 344.01
##
## Number of Fisher Scoring iterations: 8
```

No, but hard to tell, since the p-values are so high.

e) Interpret your model

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?

vg-artikkel, på måling og hva de registrerer dødsfall som. lite datasett hvem sjekker? menn mer utsatt enn damer.

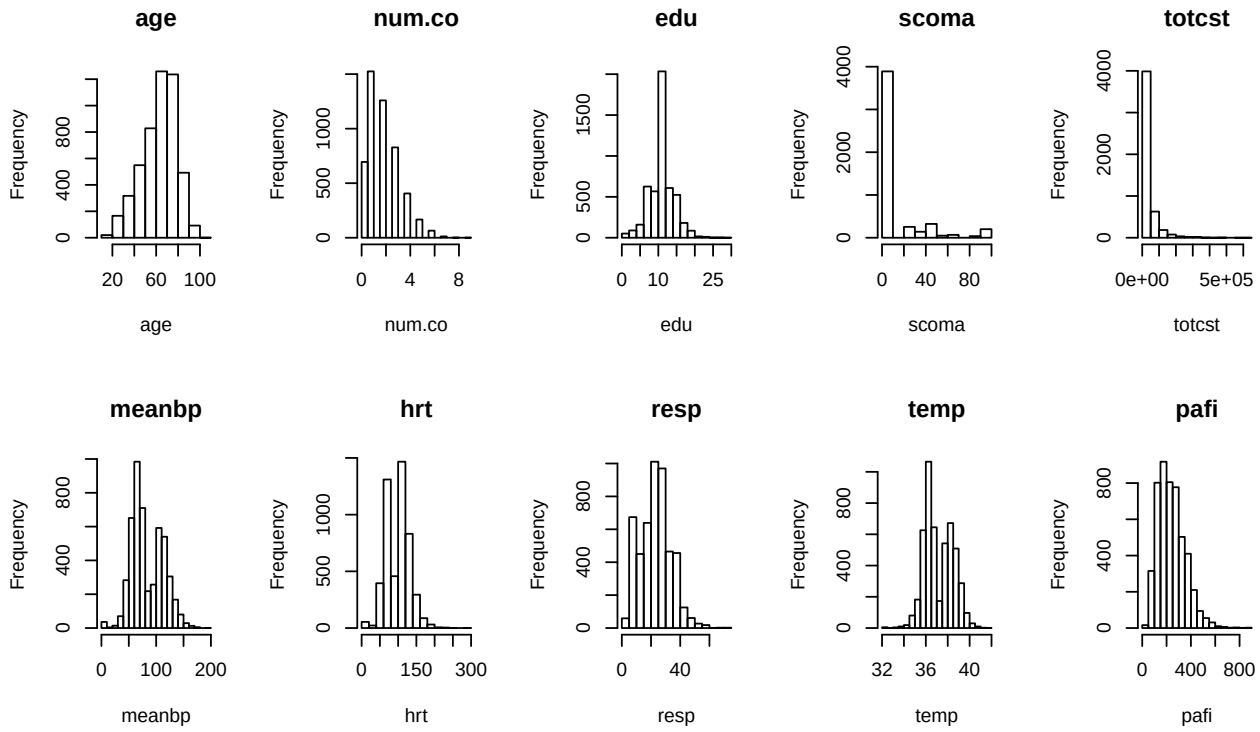
f) Multiple choice (2P)

TRUE, TRUE, FALSE, TRUE

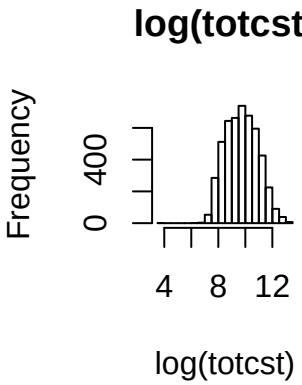
Problem 3

a)

Histograms of all integer and continuous variables are shown below.



A fitting transformation of the `totcst` variable is `log(totcst)`, as shown below.



b)

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`.

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

(i)

```
age_increase_cost <- exp(mlr_model$coefficients[2] * 10)
```

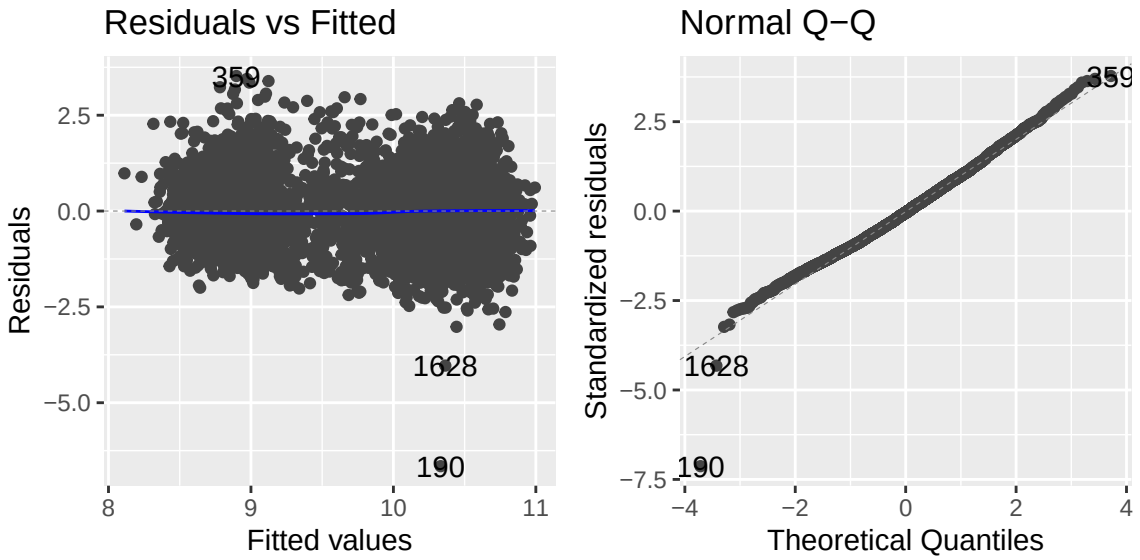
The avereage total costs increase by a factor of 0.93 when the patient's age is increased by 10 years.

exp(Beta age) = exp( ) 0.932

(ii)

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

```
autoplot(mlr_model, which = 1:2)
```



se på øving 1 eller 2 står noe om assumptions i module 2 eller 3

(iii)

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

H0 = effect of age does not depend on the diseace group HA = effect of age depends on deseace group

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

```
## 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 ***
## age             1  149.1  149.09 171.7433 < 2.2e-16 ***
## dzgroup         7 1844.0  263.43 303.4637 < 2.2e-16 ***
## age:dzgroup     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
```

Det er strong interaction p-value 2.02×10^{-4}

c)

The training and test set was created, and made into a data matrix, to use the `glmnet` package.

```
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)
```

Cross-validation was run, to find the largest λ within 1 standard error of the smallest λ.

```
ridge_model = cv.glmnet(x.train, y.train, alpha = 0)
best_lambda = ridge_model$lambda.1se
```

The value of λ was 0.142, which was then used to find the MSE of the ridge regression.

```
ridge_pred = predict(ridge_model, s = best_lambda, newx = x.test)
ridge_MSE = mean((ridge_pred - y.test)^2)
```

The final calculated MSE is 0.874.

d)

(i)

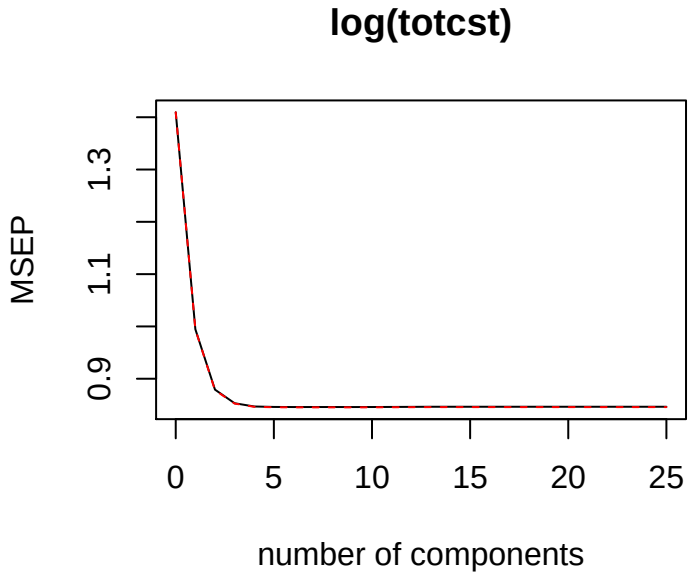
PLS regression was run, using cross-validation.

```
library(pls)
plsr_model <- plsr(log(totcst) ~ ., data = d.support.train, scale = TRUE,
                   validation = "CV")
```

(ii)

Then the validation plot was produced, to see the optinaml number of principal components.

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



The number of principal components was chosen to be 4, as this is where the curve clearly starts flattening out, and the decrease in MSE if one were to use more components is not that big. In addition, the model is simpler if we only use 4 components, instead of a higher number.

(ii)

```
plsr_predictions <- predict(plsr_model, d.support.test, ncomp = 4)
plsr_MSE <- mean((plsr_predictions - log(d.support.test$totcst))^2)
```

The final calculated MSE for PLS was 0.864. This is just slightly lower than the ridgre regression.

e)

(i)

```
gam_model <- gam(log(totcst) ~ s(age, 2) + s(temp, 6) + edu + s(resp, 7) + s(num.co,6) + dzgroup, data =
gam_pred <- predict(gam_model, newdata = d.support.test)
gam_MSE <- mean((gam_pred - y.test)^2)
```

The GAM model was fitted using different combinations of smoothing splines for the different variables, and the MSE was 0.86. This is not that impressive, but is comparable to PLS.

(i)

```
randomForest = randomForest(log(totcst) ~., data = d.support.train, mtry = ncol(d.support.train)/3, ntr
randForest_pred <- predict(randomForest, newdata = d.support.test)
randForest_MSE <- mean((randForest_pred - y.test)^2)
```

Random forest was used because it generally performs well. The MSE for the random forest was 0.824, which is by far the best MSE compared to all the other methods tested.

Problem 4 (Mixed questions)

a)

The basis functions for the cubic regression spline model is

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

, and the design matrix is given below.

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

b) Multiple choice

TRUE, TRUE, TRUE, FALSE

c) Multiple choice

FALSE, ?, TRUE, FALSE

(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.

Problem 5 (Multiple and single choice questions)

a) Multiple choice

TRUE, TRUE, FALSE, TRUE

b) Multiple choice

FALSE, TRUE, FALSE, TRUE

c) Single choice

(iv)

d) Single choice

(ii)

e) Single choice

(iii)

f) Multiple choice

TRUE, TRUE, FALSE, TRUE

g) Multiple choice

TRUE, FALSE, TRUE, TRUE