Compulsory exercise 3

TMA4268 Statistical Learning V2020

Vemund Tjessem

03 mai, 2020

Problem 1

```
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, ]
```

a)

Feature-wise normalization is performed on all the predictors for both the training and test set.

```
# Dividing training data into predictor and response and normalizing the
# predictors
x_train = college.train[, -9]
y_train = college.train[, 9]
x_train <- scale(x_train)
# Dividing test data into predictor and response and normalizing the predictors
x_test = college.test[, -9]
y_test = college.test[, 9]
x_test <- scale(x_test)
```

b)

Want to predict `Outstate` using a network with 2 hidden layers and ReLU activation functions with 64 nodes each. Will use a linear activation function for the output layer as `Outstate` is a continuous outcome. The input layer has 17 nodes. The ReLU activation function is $\phi_h(a) = \max(0,\, a)$.

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

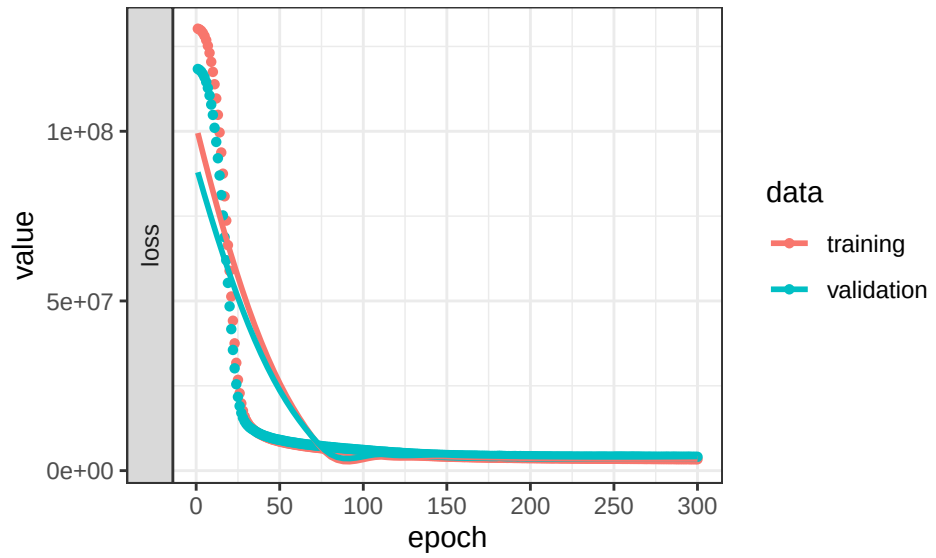
c)

(i) Train the network from b) for the training data using `keras`

```
set.seed(123)
NN.model = keras_model_sequential() %>% layer_dense(units = 64, activation = "relu",
    input_shape = c(17)) %>% layer_dense(units = 64, activation = "relu") %>% layer_dense(units = 1)
NN.model %>% compile(optimizer = "rmsprop", loss = "mse")
NN.history = NN.model %>% fit(x_train, y_train, epochs = 300, batch_size = 8, validation_split = 0.2)
```

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

```
plot(NN.history) + theme_bw()
```



The error decreases rapidly at first, but after a while the improvement for each new epoch is minimal.

(iii)

```
mse.NN.model = NN.model %>% evaluate(x_test, y_test)
```

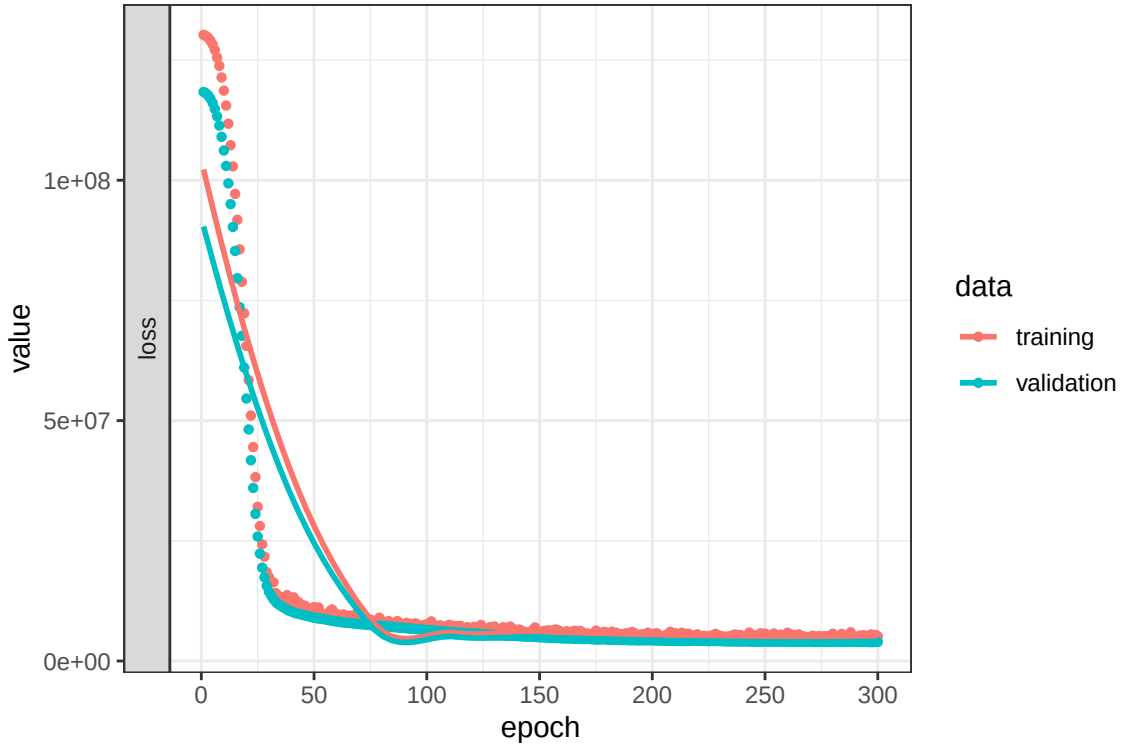
The MSE of the test set is 3.71×10^6. Compared to the methods in Compulsory 2 it performs similarly to the forward selected and lasso models which had MSEs of 3.84×10^6 and 3.9×10^6 respectively, but it is a bit better. It is not quite as good as the random forest which had a MSE of 2.57×10^6. It performed much better than all the models which used only one predictor, which obviously was expected.

d)

Applied the dropout regularization technique with a dropout rate of 0.2 for both the hidden layers as well as adding a L_2-penalty in both the hidden layers.

```
set.seed(123)
NN.model2 = keras_model_sequential() %>% layer_dense(units = 64, activation = "relu",
    input_shape = c(17), kernel_regularizer = regularizer_l2(0.001)) %>% layer_dropout(rate = 0.2) %>%
    layer_dense(units = 64, activation = "relu", kernel_regularizer = regularizer_l2(0.001)) %>%
    layer_dropout(rate = 0.2) %>% layer_dense(units = 1)
```

```
NN.model2 %>% compile(optimizer = "rmsprop", loss = "mse")
NN.history2 = NN.model2 %>% fit(x_train, y_train, epochs = 300, batch_size = 8, validation_split = 0.2)
plot(NN.history2) + theme_bw()
```




```
mse.NN.model2 = NN.model2 %>% evaluate(x_test, y_test)
```

MSE for this model is 3.5×10^6. The MSE is a bit smaller than for the previous model, but not very much so it could be down to coincidence. Even though it was better than the previous model it is not good as random forest from the previous exercise.

Problem 2

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

a)

The table showing living and deceased for each country, 0 corresponds to living and 1 corresponds to deceased.

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

```
##
```

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

Table showing living and deceased for each sex.

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

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

Table showing the number of deceased for each country, separate for each sex.

```
table(d.corona$sex[which(d.corona$deceased == 1)], d.corona$country[which(d.corona$deceased ==
    1)])
```

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

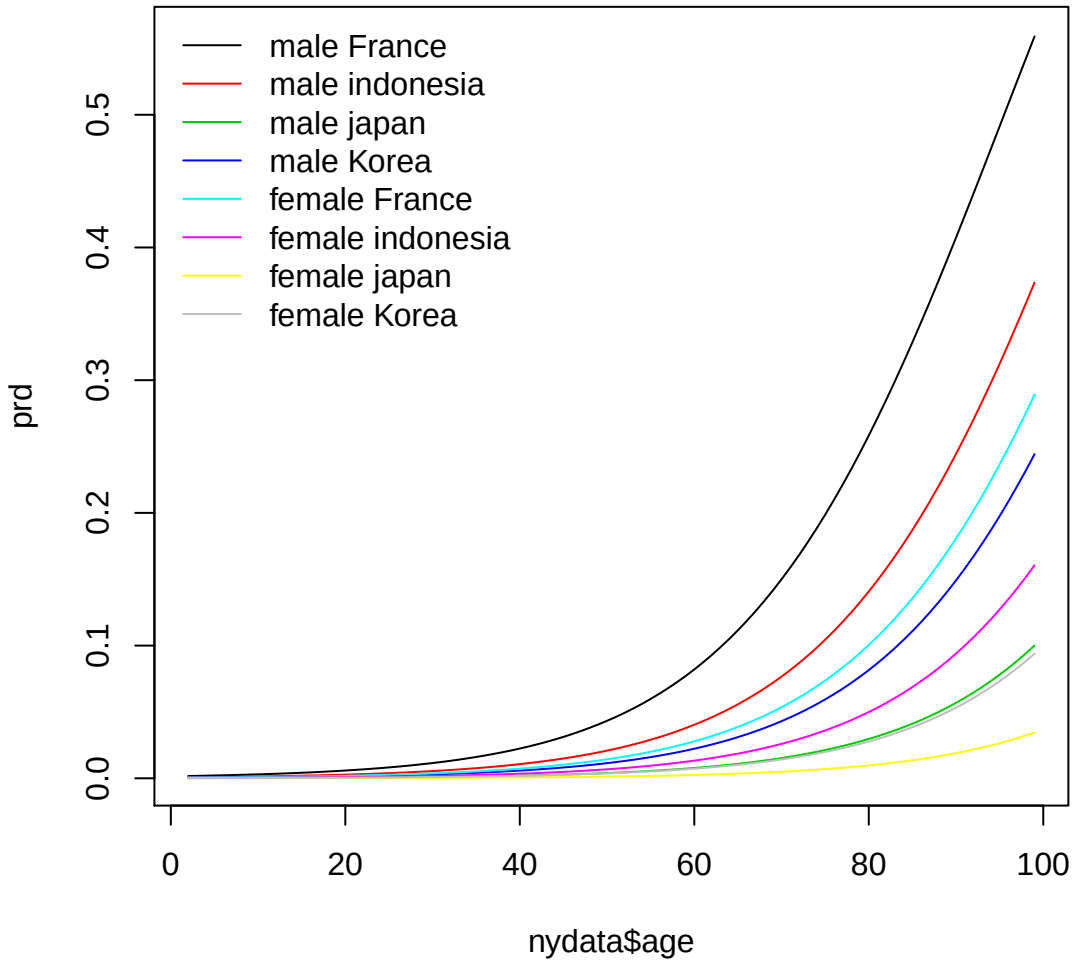
b)

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

c)

```
fit.glm = glm(deceased ~ ., data = d.corona, family = "binomial")
i = 1
for (sex_ in c("male", "female")) {
    for (country_ in c("France", "indonesia", "japan", "Korea")) {
        if (i == 1) {
            nydata = expand.grid(sex = sex_, age = seq(min(d.corona$age), max(d.corona$age),
                1), country = country_)
            prd = predict(fit.glm, nydata, type = "response")
            plot(nydata$age, prd, type = "line", col = i)
            i = i + 1
            labs = c(paste(sex_, country_))
        } else {
            nydata = expand.grid(sex = sex_, age = seq(min(d.corona$age), max(d.corona$age),
                1), country = country_)
            prd = predict(fit.glm, nydata, type = "response")
            lines(nydata$age, prd, col = i)
            i = i + 1
```

```
            labs = c(labs, paste(sex_, country_))
        }
    }
}
legend("topleft", legend = labs, col = c(1:8), lty = 1, bty = "n")
```



d)

(i)

```
glm.sex = glm(deceased ~ sex, data = d.corona, family = "binomial")
# summary(glm.sex)
coef(glm.sex)
```

```
## (Intercept)     sexmale
```

```
##  -4.3410186   0.9837844
```

Yes, as the coefficient β_{male} is positive it seems males generally have a higher probability of dying of coronavirus than females.

(ii)

```
glm.agesex = glm(deceased ~ age * sex, data = d.corona, family = "binomial")
# summary(glm.agesex)
coef(glm.agesex)
```

```
##  (Intercept)          age      sexmale  age:sexmale
## -9.280110711  0.073876575  1.386685990 -0.004067455
```

No, by looking at the coefficients, the coefficient for the interaction `age:sexmale` has a negative value which means age is not a greater risk factor for males than for females.

(iii)

```
glm.agecountry = glm(deceased ~ age * country, data = d.corona, family = "binomial")
# summary(glm.agecountry)
coef(glm.agecountry)
```

```
##          (Intercept)                   age     countryindonesia
##          -9.22100450           0.09553120           5.29256248
##         countryjapan         countryKorea age:countryindonesia
##           2.91047778           0.73700107          -0.08735423
##     age:countryjapan     age:countryKorea
##          -0.06736478          -0.02660045
```

Yes, the coefficient for `age:countryKorea` is negative, which indicates that age is a smaller risk factor for Koreans, i.e. a greater risk factor for the French population than the Korean population. The p-value is quite large though so I would not be too confident that there is much difference.

e)

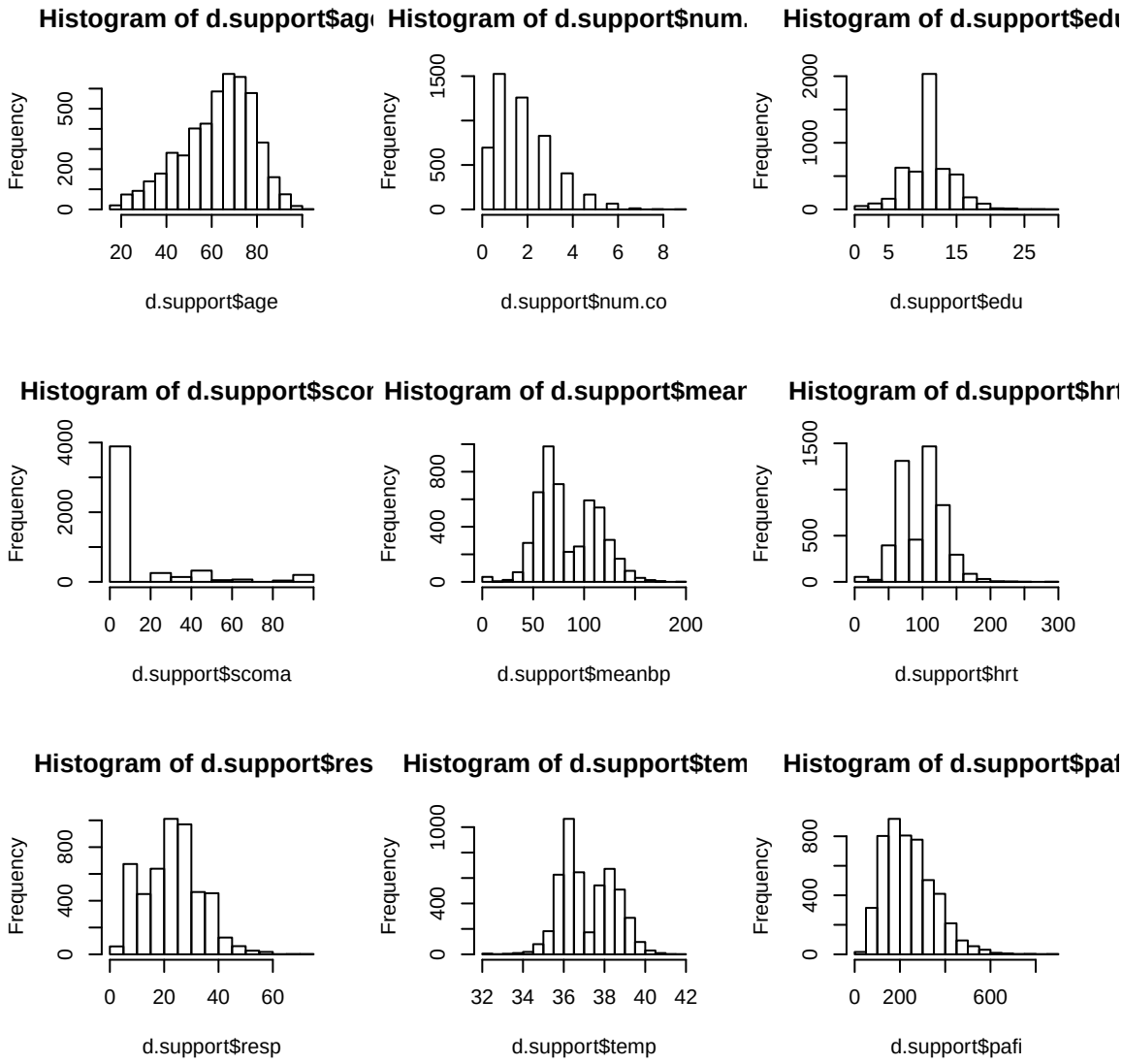
I do not trust these results. Different countries can have their own way of collecting the data which would greatly influence the results. Things such as who is tested and what counts as a death by corona virus may vary between the countries.

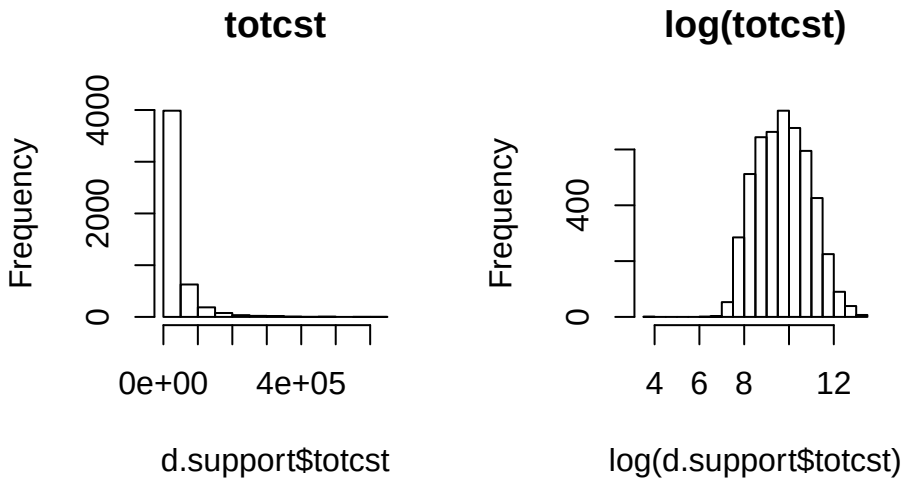
f)

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

Problem 3

a)


Histogram of d.support$ag
Frequency
0 200 500
20 40 60 80
d.support$age
Histogram of d.support$num.
Frequency
0 500 1500
0 2 4 6 8
d.support$num.co
Histogram of d.support$edu
Frequency
0 1000 2000
0 5 15 25
d.support$edu
Histogram of d.support$scor
Frequency
0 2000 4000
0 20 40 60 80
d.support$scoma
Histogram of d.support$mear
Frequency
0 400 800
0 50 100 200
d.support$meanbp
Histogram of d.support$hrt
Frequency
0 500 1500
0 100 200 300
d.support$hrt
Histogram of d.support$res
Frequency
0 400 800
0 20 40 60
d.support$resp
Histogram of d.support$tem
Frequency
0 400 1000
32 34 36 38 40 42
d.support$temp
Histogram of d.support$pafi
Frequency
0 400 800
0 200 600
d.support$pafi



Seems that the logarithm might be a fitting transformation for `totcst`.

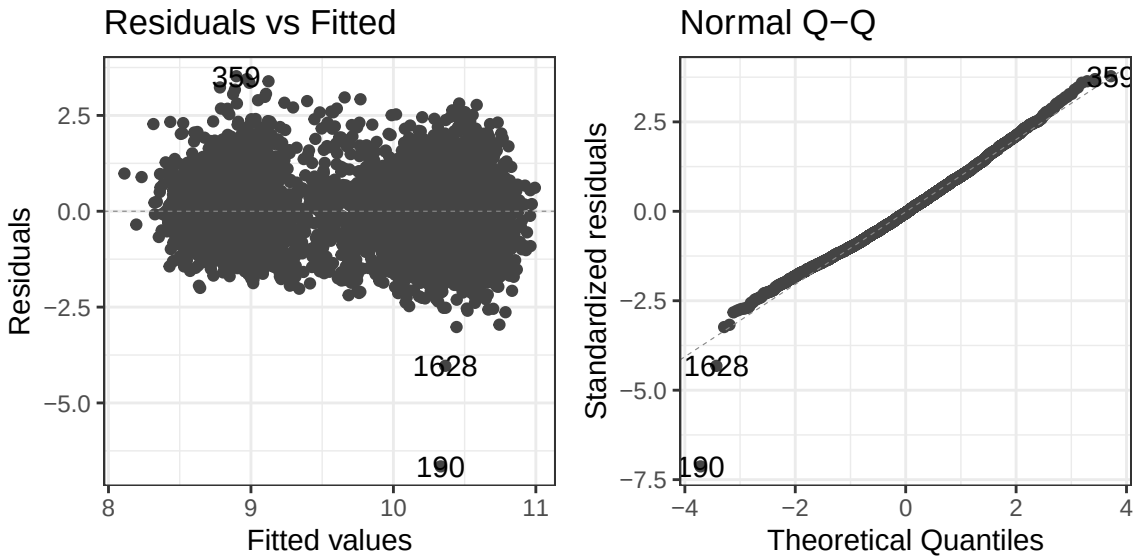
b)

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

(i) The cost changes by a factor of $\exp(\beta_{age}\, 10) = \exp(-0.006995 \cdot 10) = 0.932$.

(ii) Looking at the Tukey-Anscombe plot it seems the expected value of ϵ_i is 0 and they also seems to have the same variance. There seems to be a bit of clustering in the fitted values, so they might not be completely independent. Looking at the QQ-diagram the values seem to lie on a straight line which implies that the ϵ_i are normally distributed.

```
autoplot(mlr.model, smooth.colour = "NA")[1:2] + theme_bw()
```



(iii) Will do a test to check if the effect of age depends on the disease group

$$H_0 : \text{Effect of age does not depend on the desease group}$$
$$H_A : \text{Effect of age depends on desease group}$$

```
mlr.nointeract = lm(log(totcst) ~ age + dzgroup, data = d.support)
mlr.interact = lm(log(totcst) ~ age * dzgroup, data = d.support)
anova(mlr.nointeract, mlr.interact)
```

```
## Analysis of Variance Table
##
## Model 1: log(totcst) ~ age + dzgroup
## Model 2: log(totcst) ~ age * dzgroup
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1   4951 4409.7
## 2   4944 4384.0  7    25.744 4.1474 0.000147 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# mlr.model2 = lm(log(totcst)~temp+edu+resp+num.co+age*dzgroup, data = d.support)
# summary(mlr.model2) anova(mlr.model2)
```

There is reason to reject the null hypothesis as $Pr(> F) = 0.000147$. The effect of age depends on the desease group.

c)

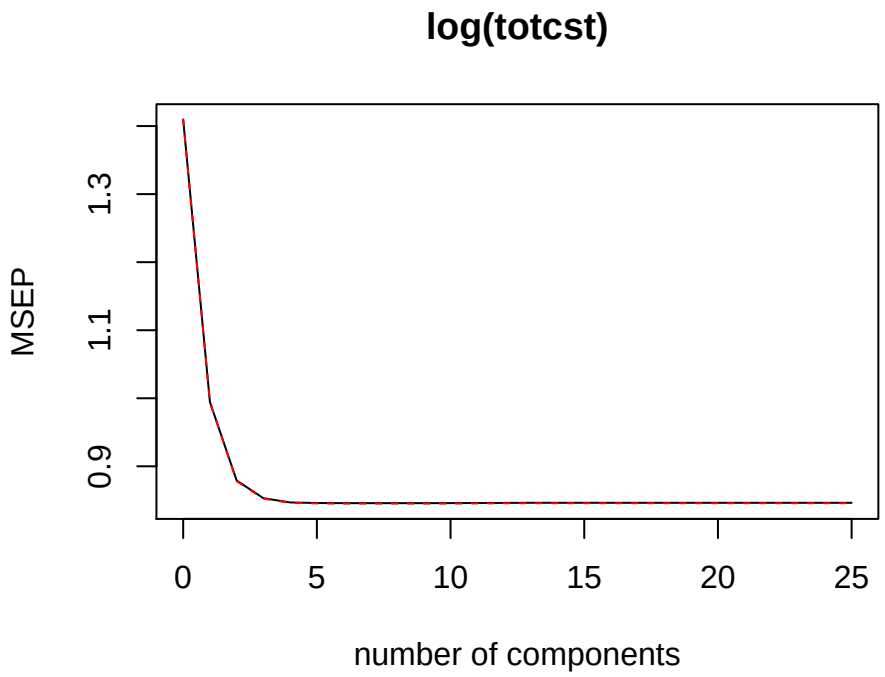
```
set.seed(12345)
train.ind = sample(1:nrow(d.support), 0.8 * nrow(d.support))
d.support.train = d.support[train.ind, ]
d.support.test = d.support[-train.ind, ]
x.train = model.matrix(totcst ~ ., data = d.support.train)[, -1]  # -1 is to remove intercept
y.train = log(d.support.train$totcst)
x.test = model.matrix(totcst ~ ., data = d.support.test)[, -1]  # -1 is to remove intercept
y.test = log(d.support.test$totcst)
ridge.fit = cv.glmnet(x.train, y.train, alpha = 0)  # alpha=0 gives ridge regression
ridge.lambda = ridge.fit$lambda.1se
ridge.pred = predict(ridge.fit, s = ridge.lambda, newx = x.test)
MSE.ridge = mean((ridge.pred - y.test)^2)
```

The value of λ was 0.142, which gave a test MSE of 0.874.

d)

(i)

```
pls.model = plsr(log(totcst) ~ ., data = d.support.train, scale = TRUE, validation = "CV")
validationplot(pls.model, val.type = "MSEP")
```




```
pls.pred = predict(pls.model, d.support.test, ncomp = 4)
MSE.pls = mean((pls.pred - y.test)^2)
```

(ii) Looking at the plot 4 components seems enough. There is minimal improvement by adding more components so a simpler model is preferred.

(iii) The MSE of the test set is 0.864. This is better than the ridge regression, but only sligthly and could be down to coincidence.

e)

(i) A GAM was fit

```
GAM.model = gam(log(totcst) ~ s(age, df = 2) + s(temp, df = 5) + s(edu, df = 2) +
    s(resp, df = 8) + s(num.co, df = 3) + dzgroup, data = d.support.train)
# plot(GAM.model, se=T) summary(GAM.model)
GAM.pred = predict(GAM.model, d.support.test)
MSE.GAM = mean(as.numeric((GAM.pred - y.test)^2))
MSE.GAM
```

```
## [1] 0.8595747
```

The MSE of the GAM was 0.86. This is better than both ridge regression and PLS.

(ii) Will use a random forest model as it generally give good results and produces more decorrelated trees than normal bagging. Will use $m = p/3$ as this is a regression tree.

```
set.seed(1)
randomForest.model = randomForest(log(totcst) ~ ., data = d.support.train, mtry = ncol(d.support.train),
    ntree = 500, importance = TRUE)
randomForest.pred = predict(randomForest.model, newdata = d.support.test)
MSE.randomForest = mean((randomForest.pred - y.test)^2)
```

The MSE on the test set is 0.824, better than all the other models.

Problem 4

a)

The basis functions are

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

The design matrix is

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

b)

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

c)

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

Problem 5

a)

(i) True

(ii) True
(iii) False
(iv) True

b)

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

c)

(iv) is true. The bias will steadily increase.

d)

(ii) is true. It means that the performance of the K-nearest neighbour classifier gest worse when the number of predictor variables p is large

e)

(iii) Is true. 0.2

f)

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

g)

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