Assignment 1

Group 2

2023-10-04

Task 1

Task 2

Task 3

Task 4

Task 5

Task 6

Task 7

```
# load data
data("diabetes", package = "lars")

# matrices can apparently be columns in df..
# good to know
y <- rnorm(100)
x1 = rnorm(100)
x2 = rnorm(100)
x <- cbind(x1, x2)
df <- data.frame(y = y, x = I(x))

# but this complicates things so we break up the
# structure and make a nice df
# function to get rid of Asis
unAsIs <- function(X) {
    if("AsIs" %in% class(X)) {
        class(X) <- class(X)[-match("AsIs", class(X))]
    }
    X
}

# extract y and x
y <- diabetes$y
x <- unAsIs(diabetes$x)
```

```r
# make a new df with all the data
diabetes_df <- as.data.frame(cbind(y, x))
```

As instructed, we now set a seed and sample row indices from the set of integers running from 1 to the number of observations with equal probability.

```r
# set seed
set.seed(12)

# split the data into train and test
# step 1: sample 400 indices
ind <- sample(x = 1:nrow(diabetes_df), size = 400)

# subset the datasets as instructed
train <- diabetes_df[ind, ]
test <- diabetes_df[-ind, ]
```

The reason why random sampling is a good idea is that we are not really familiar with the dataset. Specifically we do not know whether observations were sorted by any of the variables available and if so we do not know at all by which one. Just taking the 400 first observations then would lead the information contained in training and test data to be biased by sorting leading ultimately to sampling bias in our estimations.

INSERT EXPLANATION ABOUT STANDARDIZED VARIABLES HERE.

```r
library(kableExtra)
```

```
## Warning: Paket 'kableExtra' wurde unter R Version 4.2.2 erstellt
```

```r
# exploration of correlation
correlation_matrix <- round(cor(train), 2)
# eliminate redundancies
correlation_matrix[!lower.tri(correlation_matrix)] <- ""
kable(correlation_matrix, booktabs = T)
```

	y	age	sex	bmi	map	tc	ldl	hdl	tch	ltg	glu
y											
age	0.18										
sex	0.07	0.21									
bmi	0.6	0.17	0.09								
map	0.44	0.31	0.26	0.41							
tc	0.22	0.26	0.06	0.25	0.25						
ldl	0.19	0.23	0.16	0.26	0.21	0.9					
hdl	-0.4	-0.09	-0.39	-0.36	-0.21	0.04	-0.2				
tch	0.42	0.22	0.36	0.41	0.29	0.56	0.67	-0.74			
ltg	0.56	0.27	0.16	0.44	0.39	0.52	0.33	-0.4	0.61		
glu	0.37	0.29	0.24	0.38	0.4	0.33	0.28	-0.27	0.41	0.48	

```r
# use training data to fit the full model
# get model formula from column names
f <- as.formula(paste0("y~", paste(colnames(train)[-1], collapse = "+")))
```

```r
fit_full <- lm(data = train, formula = f)

# get variables significant at alpha = 0.05
summary_full <- summary(fit_full)
coefficients <- summary_full$coefficients
significant <- which(coefficients[, 4] < 0.05)[-1]

# in sample MSE
MSE_full_in <- mean(fit_full$residuals^2)

# out of sample MSE
pred_full <- predict(fit_full, newdata = test)
MSE_full_out <- mean((test$y - pred_full)^2)

message(paste("In sample MSE is", MSE_full_in, sep = ": "))
```

```
## In sample MSE is: 2876.35849729569
```

```r
message(paste("Out of sample MSE is", MSE_full_out, sep = ": "))
```

```
## Out of sample MSE is: 2809.87939232709
```

```r
# use the significant variables only
f2 <- as.formula(paste0("y~", paste(colnames(train)[significant], collapse = "+")))

# estimate smaller model
fit_sig <- lm(data = train, formula = f2)

# in sample MSE
MSE_sig_in <- mean(fit_sig$residuals^2)

# out of sample MSE
pred_sig <- predict(fit_sig, newdata = test)
MSE_sig_out <- mean((test$y - pred_sig)^2)

message(paste("In sample MSE is", MSE_sig_in, sep = ": "))
```

```
## In sample MSE is: 3030.1546413137
```

```r
message(paste("Out of sample MSE is", MSE_sig_out, sep = ": "))
```

```
## Out of sample MSE is: 3222.23929484251
```

```r
# F-Test
anova(fit_full, fit_sig)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ age + sex + bmi + map + tc + ldl + hdl + tch + ltg + glu
## Model 2: y ~ sex + bmi + map + ltg
```

```
##   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
## 1    389 1150543
## 2    395 1212062 -6    -61518 3.4666 0.002383 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Task 8

Task 9

Task 10