```
# ----------------------------------------------------------------------------------
# Classication and Discriminant Analysis, Wst. 2014
# Exercise 1
# Dimitrios Lenis / 9827347 / 936
# ----------------------------------------------------------------------------------



rm(list=ls())    # clean out the workspace

# import package / data
library(ElemStatLearn)
data(prostate)
attach(prostate)


trainingSet <- subset( prostate, train==TRUE, select =-train)
testSet <- subset( prostate, train==FALSE, select =-train)

# -------------------------------   1. Full model  -------------------------------

fullModel <- lm(lpsa~., data=trainingSet)
print(summary(fullModel))


# lcavol, lweight, lbph and svi yield significant t-test results, and p values
# beneath an assumed sign. a of 0.05; hence they can be used for our linear
# approximation (i.e. beta_i non zero following 3.1.4 of the script).
# The set of the remaining values should be discarded since they won't
# add significantly to an explanation for lpsa.

# R-Squared = 0.6944 i.e. the linear regression model accounts for 69.44% of
# the variance; adjusted it still accounts for 65,22%; therefore we don't have a
# particularly good fit (at least according to this value).

# Since F-Statistic = 16.47 is larger the F-Quantil = 2.10  (qf(0.95,8,58))
# the assumption H_0 doesn't hold (i.e. not a constant model).



# ----------------------------   2. Stepwise regression  ---------------------------


lm0 <- lm(lpsa~1, data=trainingSet)

lmForward <- step(lm0,  scope=formula(fullModel ), direction="forward")
lmBackward <- step(fullModel, scope=formula(lm0), direction="backward")
lmBothUp <- step(lm0, scope=formula(fullModel), direction="both")
lmBothDown <- step(fullModel, scope=formula(lm0), direction="both")

anova(lm0, lmForward, lmBothUp, lmBackward, lmBothDown, fullModel)

# Starting with the smalles model a F-Test is performed, that in return shows no
# significant gain when more than four variables (lcavol, lweight, svi and lbph)
# are used for regression


# ---------------------------   3. Best subset regression  -------------------------
```

```
library(leaps)

# ----------------------------    a. Use the best subset regression

lmSub <- regsubsets(lpsa~., data=trainingSet , nvmax=8, nbest=3)

# ----------------------------    b. Plot the results
plot(lmSub)
# a model consisting of lcavol and lweight seems to explain lpsa best, as this set
# ranks highest, while being the smallest

# ----------------------------    c. Apply lm() on the final best model
summaryLmSub <- summary(lmSub)
str(summaryLmSub)
plot(dimnames(summaryLmSub$which)[[1]], summaryLmSub$bic, main="best subset regression",
xlab="#variables", ylab="BIC")

lmBest <- lm(lpsa ~ lcavol + lweight, data=trainingSet )
summary(lmBest )

# Both lcavol and lweight yield significant t-test results, and p values
# beneath an assumed sign. a of 0.05; hence they can be used for our linear
# approximation (i.e. beta_i non zero following 3.1.4 of the script).

# R-Squared = 0.6148, i.e. the linear regression model accounts for 61.48% of
# the variance; adjusted it still accounts for 60,27%; therefore we again don't
# have a particularly good fit (at least according to this value).

# Since F-Statistic = 51.06 is larger the F-Quantil = 3.14  (qf(0.95,2,64))
# the assumption H_0 doesn't hold (i.e. not a constant model).



# -----------------  Evaluate on the test set (use MSE as a criterion)
-------------------

resTab <- matrix(nrow=6, ncol=1, dimnames=list(c("Volles Modell", "Stepwise Forward",
"Stepwise Backward", "Stepwise Forward-Backward 1", "Stepwise Forward-Backward 2", "Best-
Subset-Regression"), c("MSE")))

mse <- function(model) {mean((testSet$lpsa-predict.lm(model, newdata=testSet))^2)}

resTab[1,1] <- mse(fullModel)
resTab[2,1] <- mse(lmForward)
resTab[3,1] <- mse(lmBackward)
resTab[4,1] <- mse(lmBothUp)
resTab[5,1] <- mse(lmBothDown)
resTab[6,1] <- mse(lmBest)
print(resTab)

result <- resTab[order( resTab[,1]),][1]
print(result)

# The best fitting model, that is the one with the lowest MSE value, is the one
# resulting from the Stepwise-Forward-Regression method.
```