```
library(ElemStatLearn)
library(MASS)
library(splines)
library(ggplot2)
library(reshape2)

data(SAheart)

SAheart <- within(SAheart, rm(famhist))

set.seed(100)
train <- sample(1:nrow(SAheart), 300)

variableNames <- colnames(subset(SAheart, select = -chd))
form <- as.formula(paste("chd ~", paste(paste("ns(", variableNames, ", df = 4)", sep =
""), collapse="+")))

LRModel <- glm(form, data = SAheart[train, ], family = binomial)
summary(LRModel)

# All the basis functions of age and ldl are significant as well as selected basis
functions of sbp, tobacco and obesity.

LRPrediction <- as.numeric(predict(LRModel, newdata = SAheart[-train, ], type =
"response") > 0.5)
LRTable <- table(SAheart$chd[-train], LRPrediction)
LRRate <- 1 - sum(diag(LRTable))/sum(LRTable)

# With the particular training data, the misclassification rate is 0.327.

stepModel <- step(LRModel, direction = "both")
summary(stepModel)

# Again the basis functions of age are significant, even more so. The basis functions of
ldl became less significant,
# with the significance of the selected basis functions of obesity and sbp staying the
same. The variable tobacco
# was removed.

stepPrediction <- as.numeric(predict(stepModel, newdata = SAheart[-train, ], type =
"response") > 0.5)
stepTable <- table(SAheart$chd[-train], stepPrediction)
stepRate <- 1 - sum(diag(stepTable))/sum(stepTable)

# I was not sure how to interpret this question, but I guess the task was to plot the
variable values against the
# estimated functions. To do this, I recreated the figure 5.4 in The Elements of
Statistical Learning, Friedman et al.

stepNames <- as.character(lapply(attr(stepModel$terms, "term.labels"),  function (x)
substr(x, 4, nchar(x)-9)))

preData <- as.data.frame(lapply(SAheart[, stepNames], function(x) {
                    temp <- range(as.numeric(x))
                    return(seq(temp[1], temp[2], length.out = 500))}))

termPredictions <- predict(stepModel, preData, type = "terms", se.fit = TRUE)

ggData <- cbind(melt(preData), standarderror = melt(termPredictions$se.fit)[, "value"], y
```

```
= melt(termPredictions$fit)[, "value"])
qplot(value, y, data = ggData, geom = "line", colour = I("black"), size = I(1), xlab =
"", ylab = "") +
  geom_ribbon(aes(ymin = y - 2*standarderror, ymax = y + 2*standarderror), alpha =
I(0.5), fill = I("steelblue")) +
  geom_rug(data = melt(SAheart[, stepNames]), aes(x = value, y = NULL), colour =
I("blue"), alpha = I(0.3)) +
  facet_wrap(~ variable, nrow = 3, scale = "free")
```