```
#install.packages("ElemStatLearn")
set.seed(1337)
require(ElemStatLearn)

# data loading
data(prostate)
dtrain <- prostate[which(prostate$train==TRUE),1:9]

# part 1

#install.packages("pls")
require(pls)
# scaling is done for every segment
model.pcr <- pcr(lpsa ~ .,data=dtrain,ncomp=8, validation = "CV", segments = 10,scale =
TRUE)

summary(model.pcr)
# validation part shows the RMSEP for the best model respective to the number of
components
# the training part displays in percentage how much the variance could be explained by
the components
plot(model.pcr,plottype = "validation", val.type = "MSEP", legend = "topright")
# the model with 4 explanatory variables seems to be the optimal model, since it
descirbes a local minima

plot(dtrain[,9],model.pcr$fitted.values[,1,4],col = "green", xlab = "y", ylab = "fitted
value")
points(dtrain[,9],model.pcr$fitted.values[,1,8],col = "blue")
legend("topleft",c("4 variables","full model"),text.col=c("green","blue"))
abline(0,1)
# the plot shows that there is no big difference between the full model and the model
with the 4 variables
# this strenghens the opinion that the model with the 4 variables is the optimal model


# part 2


model.plsr <- plsr(lpsa ~ .,data=dtrain,ncomp=8, validation = "CV", segments = 10,scale =
TRUE)

summary(model.plsr)
# validation part shows the RMSEP for the best model respective to the number of
components
# the training part displays in percentage how much the variance could be explained by
the components
plot(model.plsr,plottype = "validation", val.type = "MSEP", legend = "topright")
# the model with 4 explanatory variables seems to be the optimal model, since with more
components the
# MSEP doesn't decrease much compared to with 4 variables

plot(dtrain[,9],model.plsr$fitted.values[,1,4],col = "green", xlab = "y", ylab = "fitted
value")
points(dtrain[,9],model.plsr$fitted.values[,1,8],col = "blue")
points(dtrain[,9],model.plsr$fitted.values[,1,2],col = "black")
legend("topleft",c("4 variables","full model","2
variables"),text.col=c("green","blue","black"))
abline(0,1)
# the plot shows that there is no big difference between the full model and the model
```

```
with the 4 variables
# this strenghens the opinion that the model with the 4 variables is the optimal model


#plot(model.plsr, ncomp=4,asp =1, line = TRUE)   would be another possiblity but the plot
is not that pretty

# part 3 MSE calculation

newdat<-prostate[which(prostate$train==FALSE),1:8]
truedat<-prostate[which(prostate$train==FALSE),9]
# mse function as parameters the model, data is the explanatory variable and truevalues
the real values of the test data
mse<-function(mvr,data,truevalues,ncomp){
  #yhat<-predict(mvr, comps = ncomp, data)
  mse<- mean((predict(mvr, comps = 1:ncomp, newdata = data)- truevalues)^2)
  data.frame(model = paste(deparse(substitute(mvr)),ncomp,"components"),mse)
}

rbind(mse(model.pcr,newdat,truedat,4),mse(model.pcr,newdat,truedat,8),mse(model.plsr,newda
t,truedat,2),mse(model.plsr,newdat,truedat,4),mse(model.plsr,newdat,truedat,8))

# the MSE's for the pcr models are a little bit higher than the MSE's for the two pls
models
# the lowest MSE could be observed for the PLs model with 4 variables
# the PLS model with 2 components has a lower MSE than the PCR model with 4 components
(as mentioned in the lecture the PLS method often achieves better or same results with
less components than PCR)
# for the full models the MSE are obviously equal
# so as discussed in the lecture with the model calculated by PLS-Regression better
results could be achieved compared to PCR-regression (in most cases)
```