```
# pcr2.r        James p256

library(pls)        # pcr()
library(ISLR)       # Hitters data

d0=Hitters                # 19 predictors, one response
dim(d0)                    #[1] 322    20
d1 = na.omit(d0)    # removes NAs across all cols
dim(d1)                    #[1] 263    20

# a) model with standardized predictors (full dataset)
#============================================================

set.seed(2)
m1=pcr(Salary~.,data=d1,scale=T,validation="CV")    # word data required
summary(m1)

# Data:      X dimension: 263 19
#                 Y dimension: 263 1
# Fit method: svdpc                                          # singular value decomposition
# Number of components considered: 19

# VALIDATION: RMSEP
# Cross-validated using 10 random segments.
#              (Intercept)    1 comps    2 comps    3 comps    4 comps    5 comps    6 comps
# CV                    452       348.9       352.2       353.5       352.8       350.1       349.1
# adjCV                452       348.7       351.8       352.9       352.1       349.3       348.0
#              7 comps    8 comps    9 comps    10 comps    11 comps    12 comps    13 comps
# CV             349.6       350.9       352.9        353.8        355.0        356.2        363.5
# adjCV        348.5       349.8       351.6        352.3        353.4        354.5        361.6
#              14 comps    15 comps    16 comps    17 comps    18 comps    19 comps
# CV              355.2         357.4        347.6        350.1        349.2        352.6
# adjCV         352.8         355.2        345.5        347.6        346.7        349.8

# TRAINING: % variance explained
#                1 comps    2 comps    3 comps    4 comps    5 comps    6 comps    7 comps
# X                38.31       60.16       70.84       79.03       84.29       88.63       92.26
# Salary       40.63       41.58       42.17       43.22       44.90       46.48       46.69
#                8 comps    9 comps    10 comps    11 comps    12 comps    13 comps
# X                94.96       96.28        97.26        97.98        98.65        99.15
# Salary       46.75       46.86        47.76        47.82        47.85        48.10
```

```
#              14 comps   15 comps   16 comps   17 comps   18 comps   19 comps
# X              99.47      99.75      99.89      99.97      99.99     100.00
# Salary         50.40      50.55      53.01      53.85      54.61      54.61
```

validation = CV, computes 10-fold CV errors for M components in the model

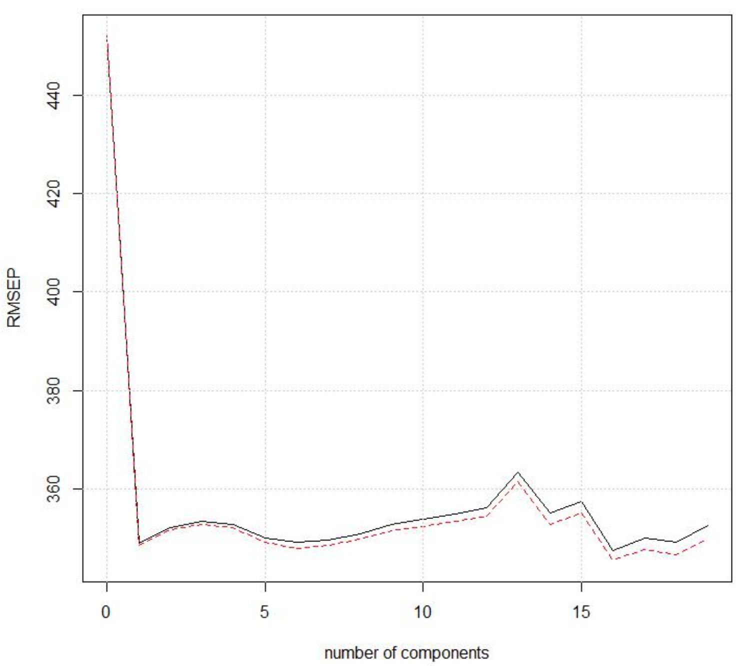
RMSEP = sqrt(MSPE)

% variance explained shows or predictors and Salary explained by model

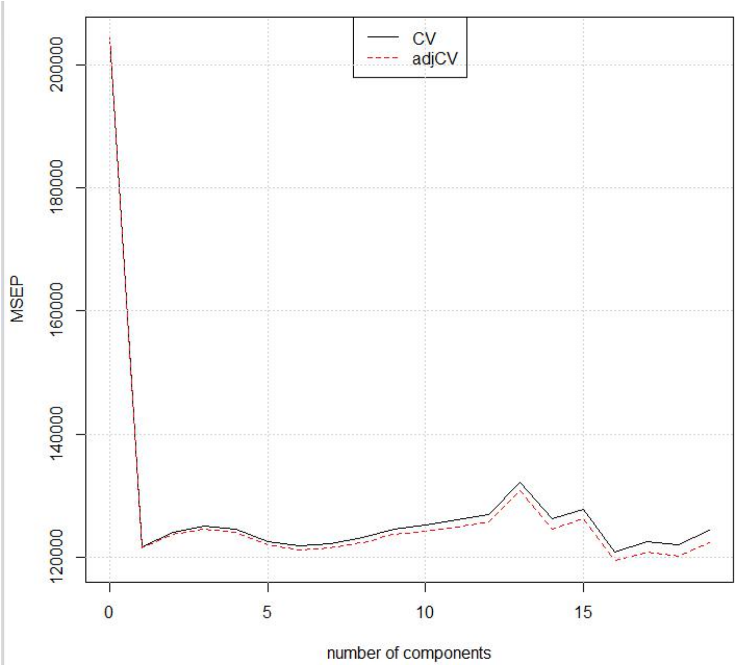
b) plot the 10-fold cv errors vs number of PCs

#==

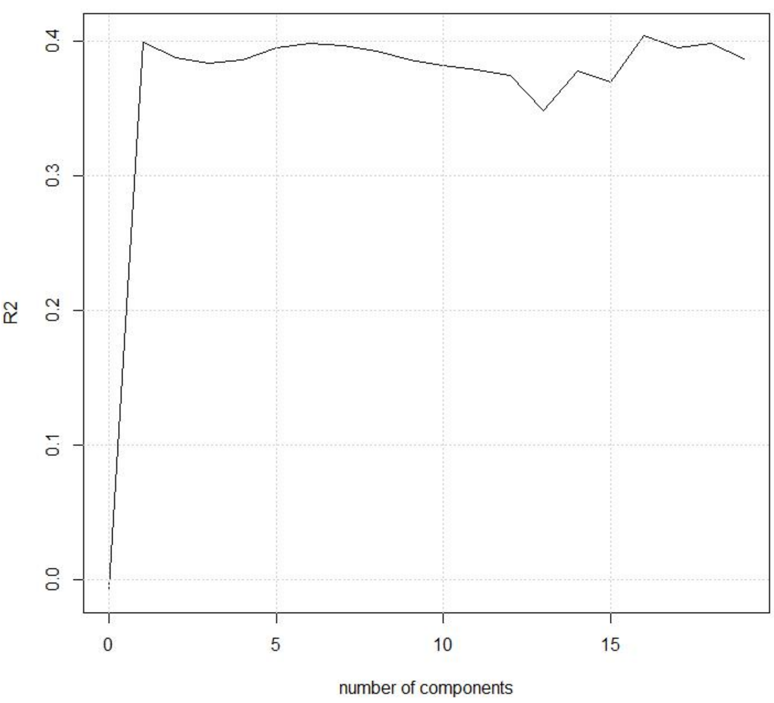
validationplot(m1,main=""); grid()



validationplot(m1,val.type="MSEP",main="",legend="top"); grid()



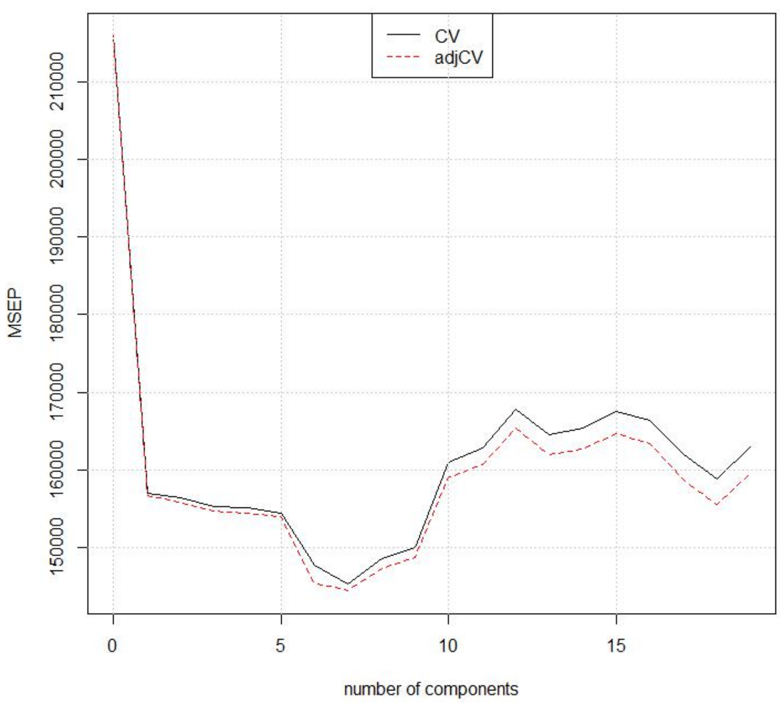
validationplot(m1,val.type = "R2",main=""); grid()



smallest CV error when M=16
However with M=1 MSPE is not much different

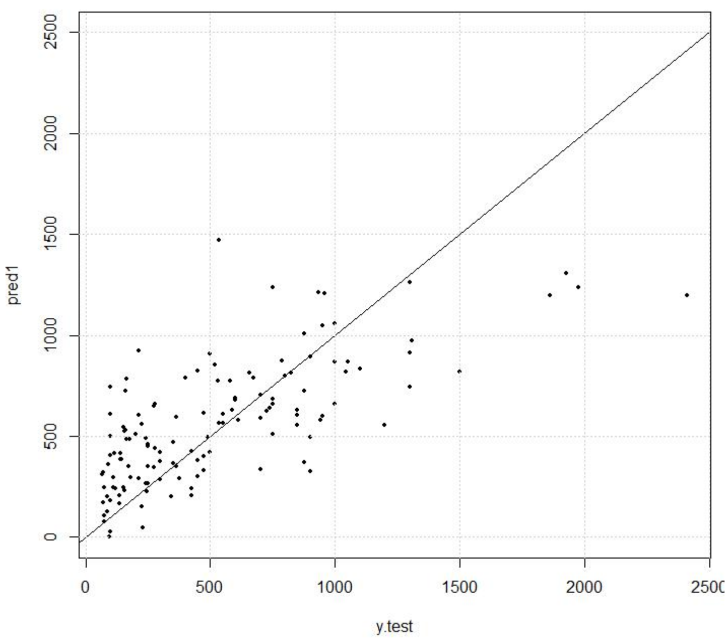
c) PCR with training set
#===
set.seed(1)

```
n=nrow(d1)
train=sample(1:n,n/2)       # 50/50
test=(-train)
dtrain=d1[train,]
dtest=d1[test,]

y=d1$Salary
y.test=y[test]

m2=pcr(Salary~.,data=dtrain,scale=T, validation="CV")      # word data required

# training plot
validationplot(m2,val.type="MSEP",main="",legendpos="top"); grid()
# lowest CV error when M=7 principal components
```



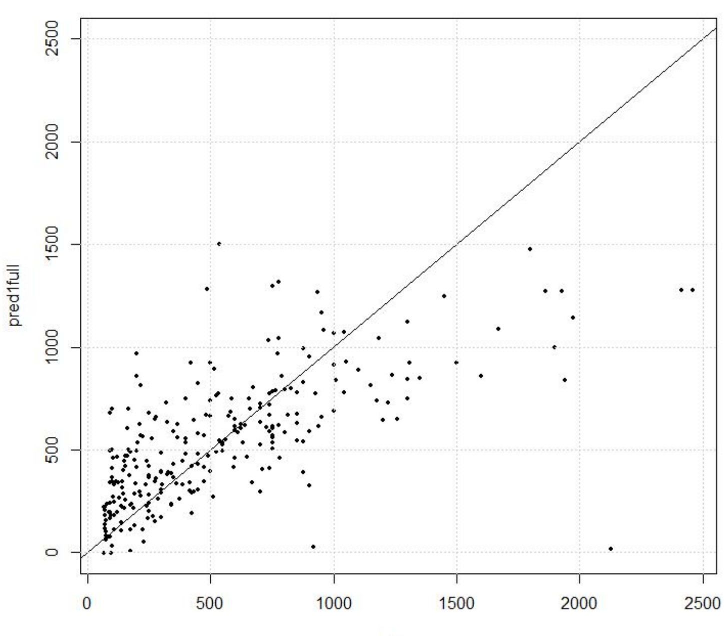

```
# test MSE
newval = dtest[,-19]                                   # needed?
pred1=predict(m2,newval,ncomp=7)
cvk0 = mean((pred1-y.test)^2)                          #[1] 96556.22

# compare predictions vs Salary
head(dtest)[,19:20]
#                         Salary NewLeague
#-Alan Ashby                475          N
#-Andre Dawson              500          N
```

```
#-Alfredo Griffin       750               A
#-Argenis Salazar       100               A
#-Andres Thomas          75               N
#-Andre Thornton       1100               A

head(y.test)     # [1]   475   500   750   100    75 1100
head(pred1)      # [1] 613.61332 906.59077 509.76018   28.21284 108.83181 831.63034

# plot prediction performance of model with 7 PCs
plot(pred1~y.test,pch=19,cex=0.6,ylim=c(0,2500))
abline(0,1)
grid()
```




```
# predicting a single obs
newval = dtrain[,-19]
newval[1,] = data.frame(AtBat=315 ,Hits= 99, HmRun=22 , Runs=33 , RBI=44 , Walks=33 , Years=11 , CAtBat=3000 , CHits=999 , CHmRun=77 , CRuns=344 , CRBI=233 , CWalks=333 , League="N" , Division="W" , PutOuts=444 , Assists=55 , Errors=11 ,NewLeague= "A")
newval = newval[1,]
pred1=predict(m2,newval,ncomp=7)
#                              Salary
#-Darryl Strawberry 484.6758


# fit PCR on full data set using M=7
```

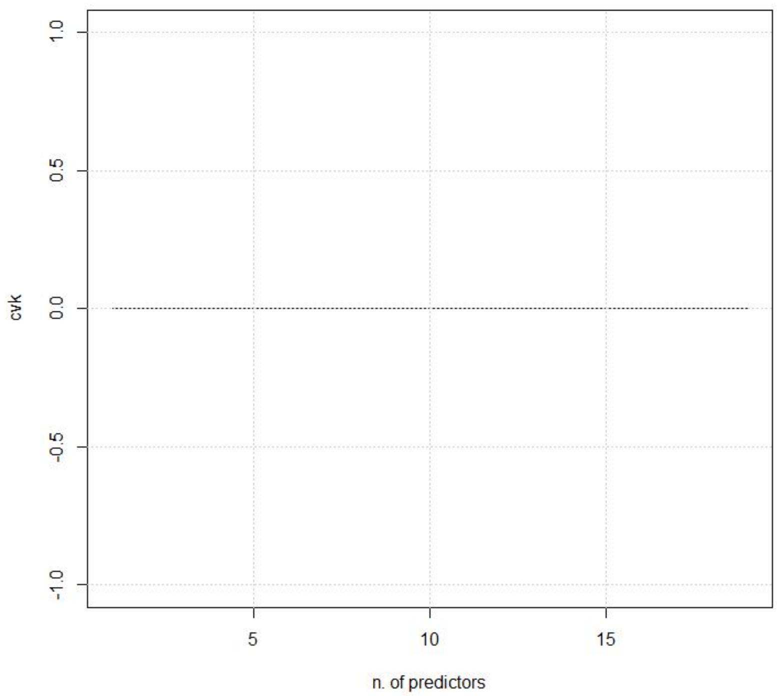
```
#=============================================================

m3=pcr(Salary~.,data=d1,scale=T,ncomp=7)
summary(m3)
#Data:      X dimension: 263 19
#           Y dimension: 263 1
#Fit method: svdpc
#Number of components considered: 7
#TRAINING: % variance explained
#     1 comps   2 comps   3 comps   4 comps   5 comps   6 comps   7 comps
#X     38.31     60.16     70.84     79.03     84.29     88.63     92.26
#y     40.63     41.58     42.17     43.22     44.90     46.48     46.69

# test MSE
newval = d1[,-19]
pred1full=predict(m3,newval,ncomp=7)
cvk0full = mean((pred1full-y)^2)                    #[1] 108084.9

# plot prediction performance of model with 7 PCs
plot(pred1full~y,pch=19,cex=0.6,ylim=c(0,2500))
abline(0,1)
grid()
```




```
# 10-fold CV on MLR
# ==================================================
# load predict.regsubsets() function
```

```
library(leaps)              # regsubsets()
n <- nrow(d1)                 # [1] 263
k <- 10                         # set the number of folds equal to 5
set.seed(1)                    # set for reproducible results

# test sets
folds <- sample(1:k,size = n,replace=T)
# vector with nums 1 to 10 (which rows belong to each of 10 folds)
folds[1:22]
# 3   4   6 10   3   9 10   7   7   1   3   2   7   4   8   5   8 10   4   8 10   3

table(folds)
#   1   2   3   4   5   6   7   8   9 10
# 13 25 31 32 33 27 26 30 22 24       # sizes of test sets, not same in all folds

mspe <- matrix(0, k, 19)                   # matrix of 0s
dim(mspe)       # [1] 10 19
# mspe[j,i] = MSPE of best model with i predictors using jth fold
for(j in 1:k)      # loop folds
{
   y = d1$Salary[folds == j]          # y-values in test set
   d2 = d1[folds != j,]                   # training set
   best.fit <- regsubsets(Salary ~.,d2,nvmax=19)
   for(i in 1:19)                             # i number of predictors in model
   {
      newdata = d1[folds ==j,]      # test set
      yhat <- predict.regsubsets(best.fit,newdata,id=i)
      mspe[j, i] <- mean((y - yhat)^2)
   }
}
mspe[,1:7]
#                  [,1]          [,2]          [,3]          [,4]          [,5]          [,6]          [,7]
# [1,] 187479.08 141652.61 163000.36 169584.40 141745.39 151086.36 193584.17
# [2,]   96953.41   63783.33   85037.65   76643.17   64943.58   56414.96   63233.49
# [3,] 165455.17 167628.28 166950.43 152446.17 156473.24 135551.12 137609.30
# [4,] 124448.91 110672.67 107993.98 113989.64 108523.54   92925.54 104522.24
# [5,] 136168.29   79595.09   86881.88   94404.06   89153.27   83111.09   86412.18
# [6,] 171886.20 120892.96 120879.58 106957.31 100767.73   89494.38   94093.52
# [7,]   56375.90   74835.19   72726.96   59493.96   64024.85   59914.20   62942.94
# [8,]   93744.51   85579.47   98227.05 109847.35 100709.25   88934.97   90779.58
```

```
# [9,] 421669.62 454728.90 437024.28 419721.20 427986.39 401473.33 396247.58
#[10,] 146753.76 102599.22 192447.51 208506.12 214085.78 224120.38 214037.26

# rows are folds
# mspe on i fold when using best model with j predictors

cvk <- apply(mspe, 2, mean)
#  [1]  160093.5  140196.8  153117.0  151159.3  146841.3  138302.6  144346.2  130207.7  129459.6  125334.7  125153.8
128273.5 133461.0 133974.6 131825.7 131882.8
#[17] 132750.9 133096.2 132804.7
plot(cvk,type="l",xlab="n. of predictors")
grid()
```




```
aux = which.min(cvk)      # 11
cvk[11]                         # 125153.8
sqrt(cvk[11])                 # 353.7709
# compare to RMSEP = 348.9 with PC1 only.

m3 = regsubsets(Salary ~.,d1,nvmax=19)
coef(m3,aux)
#    (Intercept)            AtBat             Hits            Walks           CAtBat            CRuns
#    135.7512195      -2.1277482        6.9236994        5.6202755      -0.1389914        1.4553310
#               CRBI           CWalks          LeagueN       DivisionW          PutOuts          Assists
#        0.7852528      -0.8228559       43.1116152 -111.1460252        0.2894087        0.2688277
```