Regression with principal components in R

Data Mining
Master Degree in Computer Science
University of Padova

a.y. 2017/2018

Annamaria Guolo

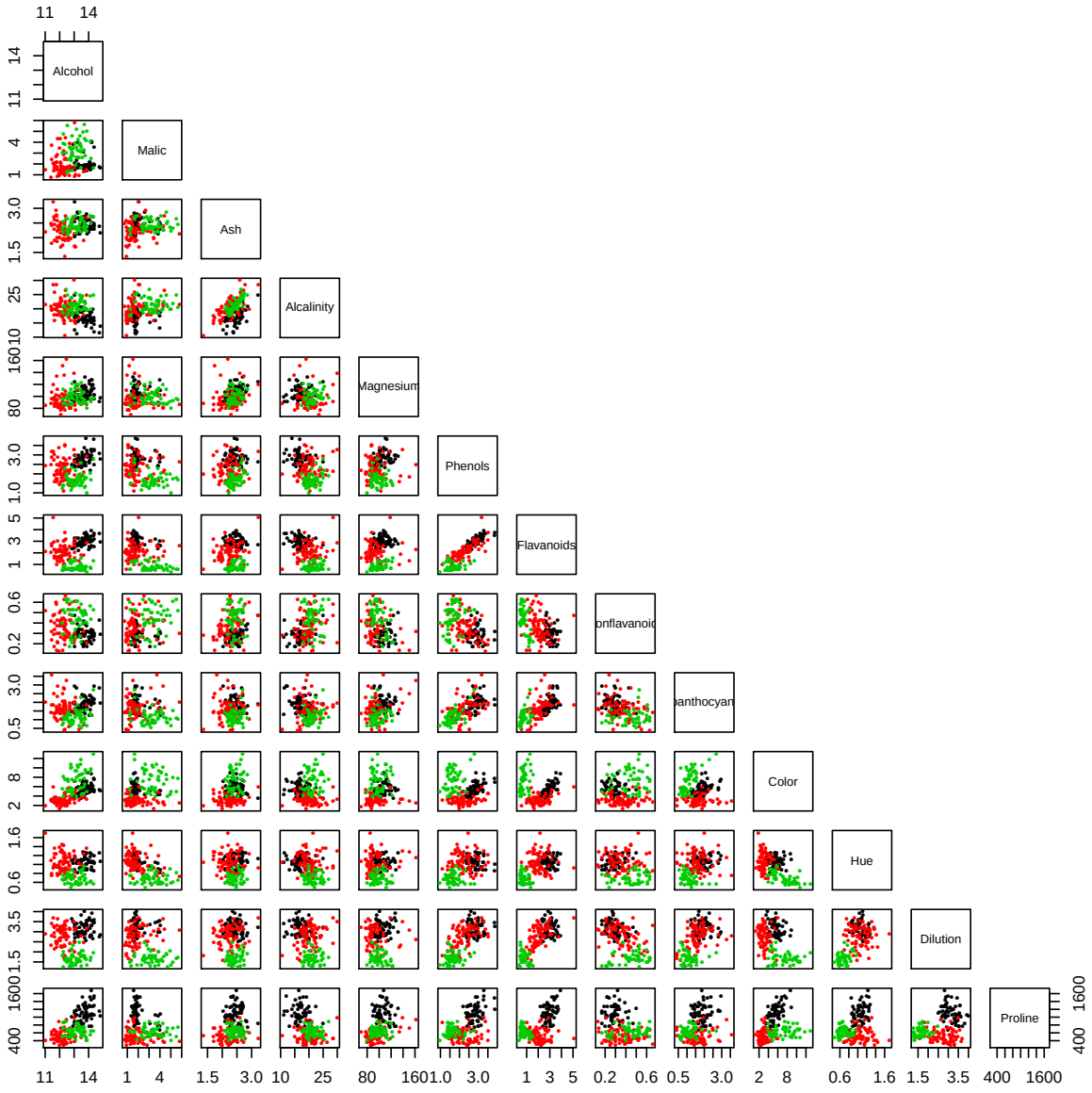
1 Wine dataset

Consider the `wine` dataset already investigated using the discriminant analysis. The following analysis is partly based on the results published on the R-Bloggers website. Data refer to a chemical analysis of different types of wives from three (*cultivar*). Thirteen chemicals are examined.

```
data(wine, package='rattle.data')
dim(wine)

## [1] 178  14

names(wine)

##  [1] "Type"            "Alcohol"         "Malic"           "Ash"
##  [5] "Alcalinity"      "Magnesium"       "Phenols"         "Flavanoids"
##  [9] "Nonflavanoids"   "Proanthocyanins" "Color"           "Hue"
## [13] "Dilution"        "Proline"
```

Plots of the relationship between variables

```
pairs(wine[,-1], col = wine$Type, upper.panel = NULL, pch = 16, cex = 0.5)
legend("topright", bty = "n", legend = c("tipo 1","tipo 2","tipo 3"),
       pch = 16, col = c("black","red","green"))
```



We can evaluate whether the PC analysis can help in intepreting the relationships among the 13 chemicals.

```
pr <- prcomp(wine[,-1], scale=TRUE)
```

Function `prcomp` perform the PC analysis: option `scale=TRUE` is needed to scale the variables.
Object `pr` contains

```
names(pr)

## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

- `sdev`: the square root of the variance explained by each PC
- `rotation`: loading vectors
- `center`: mean of the variables
- `scale`: scale of the variables
- `x`: matrix whose columns are the scores for all the units

The first two quantities appear when calling the object (not reported here for reason of space)

```
pr
```

We access the quantities as follows

```
pr$center
```

```
##         Alcohol           Malic             Ash      Alcalinity       Magnesium
##      13.0006180       2.3363483       2.3665169      19.4949438      99.7415730
##         Phenols      Flavanoids   Nonflavanoids Proanthocyanins           Color
##       2.2951124       2.0292697       0.3618539       1.5908989       5.0580899
##             Hue        Dilution         Proline
##       0.9574494       2.6116854     746.8932584
```

```
pr$scale
```

```
##         Alcohol           Malic             Ash      Alcalinity       Magnesium
##       0.8118265       1.1171461       0.2743440       3.3395638      14.2824835
##         Phenols      Flavanoids   Nonflavanoids Proanthocyanins           Color
##       0.6258510       0.9988587       0.1244533       0.5723589       2.3182859
##             Hue        Dilution         Proline
##       0.2285716       0.7099904     314.9074743
```

```
dim(pr$x)
```

```
## [1] 178  13
```

Loading vectors

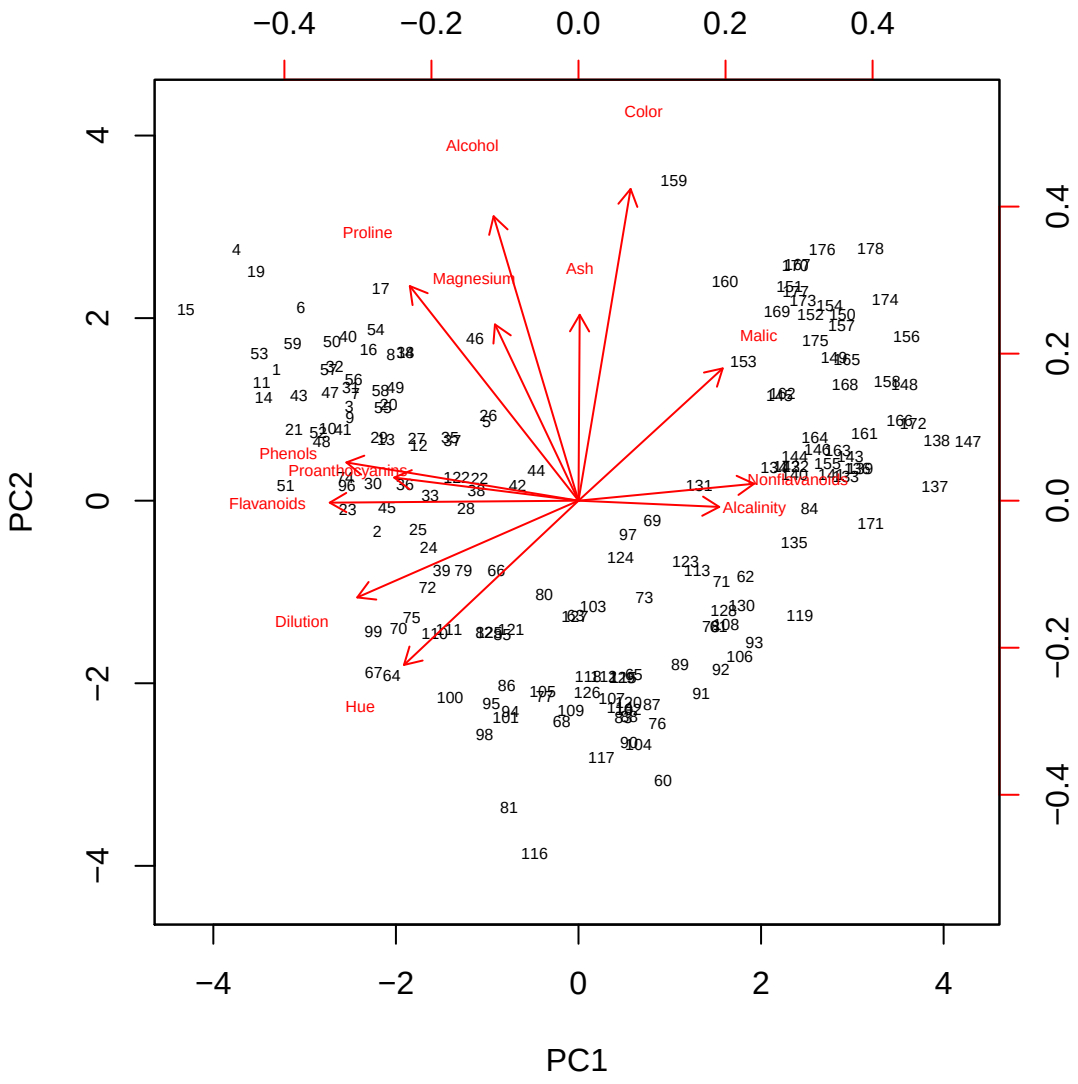
```
pr$rotation
```

```
##                          PC1          PC2         PC3         PC4         PC5         PC6
## Alcohol         -0.144329395  0.483651548 -0.20738262  0.01785630 -0.26566365  0.21353865
## Malic            0.245187580  0.224930935  0.08901289 -0.53689028  0.03521363  0.53681385
## Ash              0.002051061  0.316068814  0.62622390  0.21417556 -0.14302547  0.15447466
```

```
## Alcalinity       0.239320405 -0.010590502  0.61208035 -0.06085941  0.06610294 -0.10082451
## Magnesium       -0.141992042  0.299634003  0.13075693  0.35179658  0.72704851  0.03814394
## Phenols         -0.394660845  0.065039512  0.14617896 -0.19806835 -0.14931841 -0.08412230
## Flavanoids      -0.422934297 -0.003359812  0.15068190 -0.15229479 -0.10902584 -0.01892002
## Nonflavanoids    0.298533103  0.028779488  0.17036816  0.20330102 -0.50070298 -0.25859401
## Proanthocyanins -0.313429488  0.039301722  0.14945431 -0.39905653  0.13685982 -0.53379539
## Color            0.088616705  0.529995672 -0.13730621 -0.06592568 -0.07643678 -0.41864414
## Hue             -0.296714564 -0.279235148  0.08522192  0.42777141 -0.17361452  0.10598274
## Dilution        -0.376167411 -0.164496193  0.16600459 -0.18412074 -0.10116099  0.26585107
## Proline         -0.286752227  0.364902832 -0.12674592  0.23207086 -0.15786880  0.11972557
##                         PC7         PC8         PC9        PC10        PC11        PC12
## Alcohol         -0.05639636  0.39613926 -0.50861912  0.21160473  0.22591696 -0.26628645
## Malic            0.42052391  0.06582674  0.07528304 -0.30907994 -0.07648554  0.12169604
## Ash             -0.14917061 -0.17026002  0.30769445 -0.02712539  0.49869142 -0.04962237
## Alcalinity      -0.28696914  0.42797018 -0.20044931  0.05279942 -0.47931378 -0.05574287
## Magnesium        0.32288330 -0.15636143 -0.27140257  0.06787022 -0.07128891  0.06222011
## Phenols         -0.02792498 -0.40593409 -0.28603452 -0.32013135 -0.30434119 -0.30388245
## Flavanoids      -0.06068521 -0.18724536 -0.04957849 -0.16315051  0.02569409 -0.04289883
## Nonflavanoids    0.59544729 -0.23328465 -0.19550132  0.21553507 -0.11689586  0.04235219
## Proanthocyanins  0.37213935  0.36822675  0.20914487  0.13418390  0.23736257 -0.09555303
## Color           -0.22771214 -0.03379692 -0.05621752 -0.29077518 -0.03183880  0.60422163
## Hue              0.23207564  0.43662362 -0.08582839 -0.52239889  0.04821201  0.25921400
## Dilution        -0.04476370 -0.07810789 -0.13722690  0.52370587 -0.04642330  0.60095872
## Proline          0.07680450  0.12002267  0.57578611  0.16211600 -0.53926983 -0.07940162
##                        PC13
## Alcohol          0.01496997
## Malic            0.02596375
## Ash             -0.14121803
## Alcalinity       0.09168285
## Magnesium        0.05677422
## Phenols         -0.46390791
## Flavanoids       0.83225706
## Nonflavanoids    0.11403985
## Proanthocyanins -0.11691707
## Color           -0.01199280
## Hue             -0.08988884
## Dilution        -0.15671813
## Proline          0.01444734
```

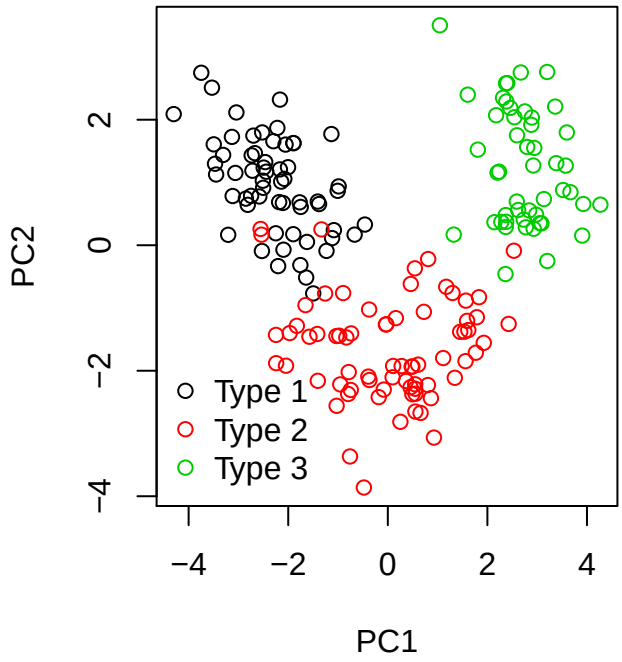
There 13 PCsC: in general, from a dataset with n rows and p columns we can obtain $min(n-1, p)$ PCs.
Results for PC1 and PC2

```
biplot(pr, scale=0, cex=0.5)
```



The plot shows the values of the scores as numbers associated to each observation in the dataset.
Option `scale=0` indicates that covariates need to be scaled.
What can we read from the plot?
Plot the points as scores associated to PC1 and PC2, by distinguishing the type of wine,

```
plot(pr$x[,1:2], col=wine$Type)
legend('bottomleft', pch=c(1,1,1), col=c(1,2,3),
       legend=c('Type 1', 'Type 2', 'Type 3'), bty='n')
```



PC1 and PC2 provide a satisfactory separation of the observations.
Compute the amount of variance explained by the PCs

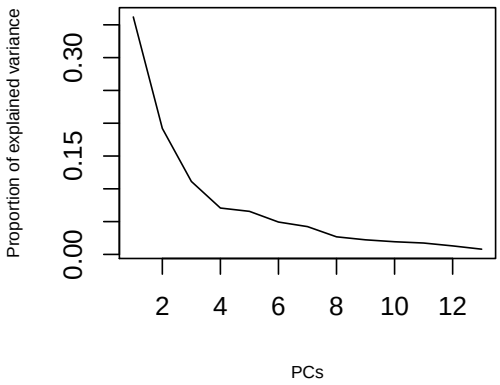
```
variance <- pr$sdev^2
```

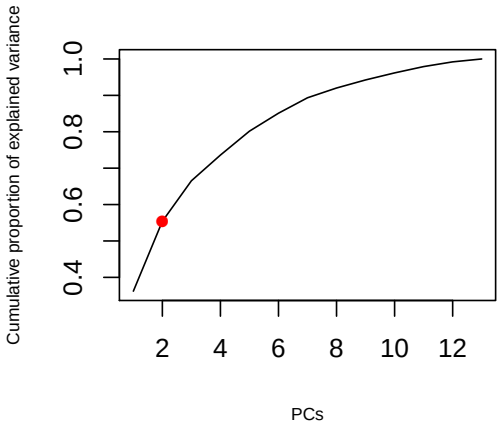
As a proportion of the total variance

```
prop.variance <- variance/sum(variance)
prop.variance

##  [1] 0.361988481 0.192074903 0.111236305 0.070690302 0.065632937 0.049358233 0.042386793
##  [8] 0.026807489 0.022221534 0.019300191 0.017368357 0.012982326 0.007952149
```

Plot the proportion and the cumulative proportion of variance explained by the PCs

```
par(mfrow=c(1,2))
plot(prop.variance, xlab='PCs',
       ylab='Proportion of explained variance', type='l', cex.lab=0.7)
## function cumsum computs the cumulative sum
plot(cumsum(prop.variance), xlab='PCs',
       ylab='Cumulative proportion of explained variance', type='l', cex.lab=0.7)
## add the point corresponding to the first two PCs
points(2, cumsum(prop.variance)[2], col=2, pch=16)
```





2 Gasoline dataset

Consider the dataset `gasoline` in library `pls`. Data refer to NIR spectra[1] and octane numbers of 60 gasoline samples.

```
library(pls)
data(gasoline)
names(gasoline)

## [1] "octane" "NIR"

## gasoline contains two objects, a vector (Y=octane) and a matrix (X=NIR)
dim(gasoline$NIR)

## [1]  60 401
```

The covariates included in NIR are 401. Start the PC regression by choosing the number M of PC through cross validation. For simplicity create the following objects

```
y <- gasoline$octane
X <- gasoline$NIR
```

```
set.seed(222)
m.pcr <- pcr(y ~ X, ncomp=20, scale=TRUE, validation='CV')
```

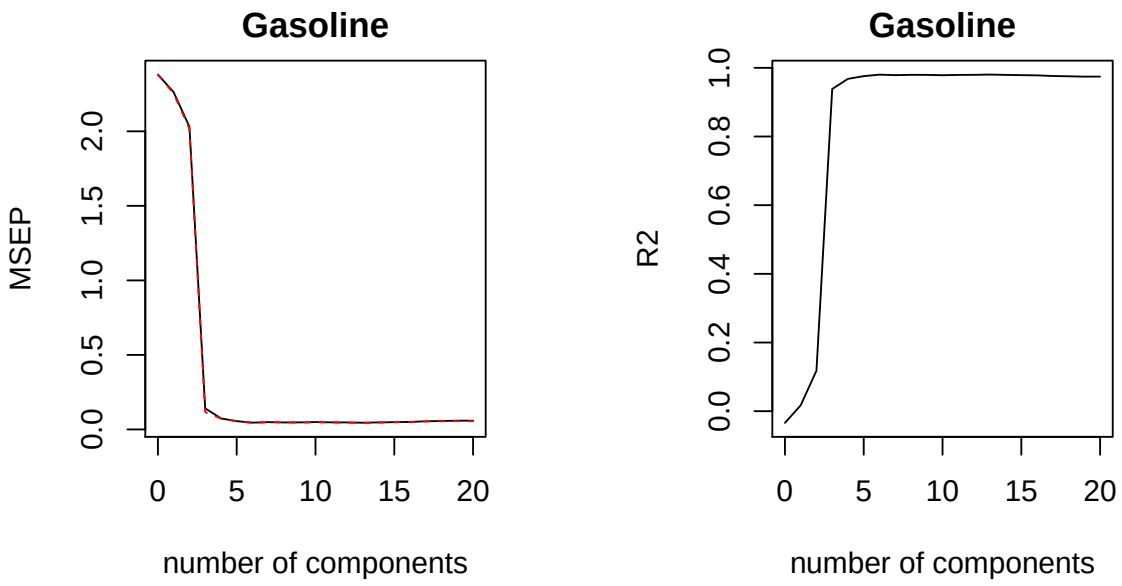
Function `pcr()` specifies the relationship between `y` and `X` using the standard syntax in `lm()`. Specification `ncomp=20` fixes the maximum number of PCs to consider in the computation. Specification `scale=TRUE` scales the variables. Specification `validation=CV` indicates to use cross validation, 10-folds cross validation by default.

[1] Spettroscopia nel vicino infrarosso: analisi che usa la regione infrarossa dello spettro elettromagnetico per studiare in modo non distruttivo le proprietà chimico-fisiche dei campioni

```
summary(m.pcr)

## Data:  X dimension: 60 401
##  Y dimension: 60 1
## Fit method: svdpc
## Number of components considered: 20
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## CV           1.543    1.504    1.425   0.3765   0.2718   0.2357   0.2135   0.2211
## adjCV        1.543    1.501    1.420   0.3410   0.2700   0.2338   0.2092   0.2193
##        8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## CV      0.2167   0.2172    0.2224    0.2180    0.2168    0.2124    0.2174    0.2211
## adjCV   0.2174   0.2138    0.2189    0.2152    0.2129    0.2094    0.2142    0.2184
##        16 comps  17 comps  18 comps  19 comps  20 comps
## CV       0.2245    0.2342    0.2378    0.2429    0.2421
## adjCV    0.2209    0.2313    0.2335    0.2373    0.2360
##
## TRAINING: % variance explained
##    1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps
## X   71.725    88.57    93.74    97.51    98.28    98.67    99.01    99.20    99.36
## y    8.856    22.69    96.39    97.40    98.18    98.51    98.51    98.57    98.79
##    10 comps  11 comps  12 comps  13 comps  14 comps  15 comps  16 comps  17 comps
## X     99.48     99.57     99.64     99.70     99.74     99.78     99.81     99.83
## y     98.79     98.81     98.88     98.88     98.88     98.88     98.93     98.93
##    18 comps  19 comps  20 comps
## X     99.85     99.86     99.88
## y     99.00     99.05     99.08
```

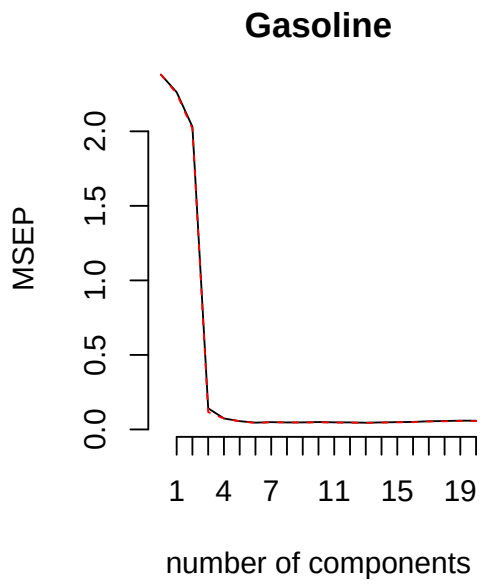
The output provides the result of the cross validation in terms of square root of the MSE for each number of PCs, until the specified maximum equal to 20. Choose the optimum through a graphical inspection of the results. Function `validationplot()` plots the values of MSEP (P=predictive) and R^2.

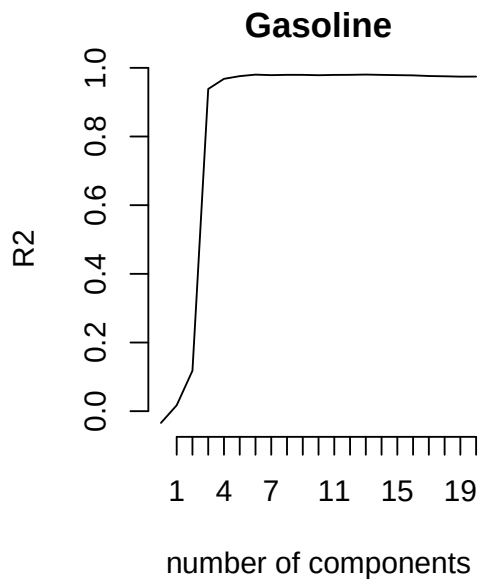
```
par(mfrow=c(1,2))
validationplot(m.pcr, val.type='MSEP', main='Gasoline')
validationplot(m.pcr, val.type='R2', main='Gasoline')
```



The same plot with some graphical amelioration on the x-axis

```
par(mfrow=c(1,2))
## graph without axes
validationplot(m.pcr, val.type='MSEP', main='Gasoline', axes=FALSE)
## add on the x-axis (1) with the specification (at) of the points at which tick-marks
## are to be drawn
axis(1, at=1:20)
## add on the y-axis
axis(2)
validationplot(m.pcr, val.type='R2', main='Gasoline', axes=FALSE)
axis(1, at=1:20)
axis(2)
```





Graphically, it seems that 4, 5 PCs are sufficient. Consider a formal evaluation

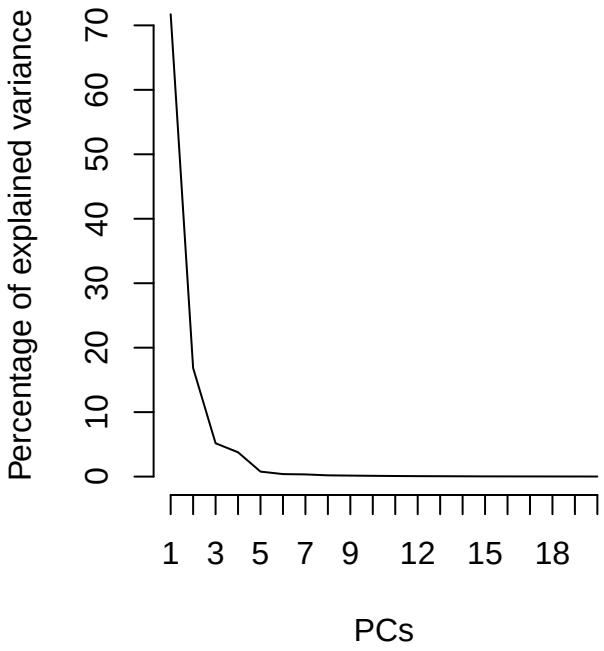
```
selectNcomp(m.pcr, method='onesigma', ncomp=20)

## [1] 5
```

We choose 5 PCs. How much variance is explained?

```
explvar(m.pcr)

##      Comp 1      Comp 2      Comp 3      Comp 4      Comp 5      Comp 6      Comp 7
## 71.72466749 16.84355942  5.16969875  3.77274681  0.77158999  0.38790757  0.33858298
##      Comp 8      Comp 9     Comp 10     Comp 11     Comp 12     Comp 13     Comp 14
##  0.19445609  0.15867287  0.11934739  0.09112767  0.06904035  0.05728967  0.04483529
##     Comp 15     Comp 16     Comp 17     Comp 18     Comp 19     Comp 20
##  0.03255193  0.02972121  0.02271663  0.01849377  0.01743618  0.01328488
```

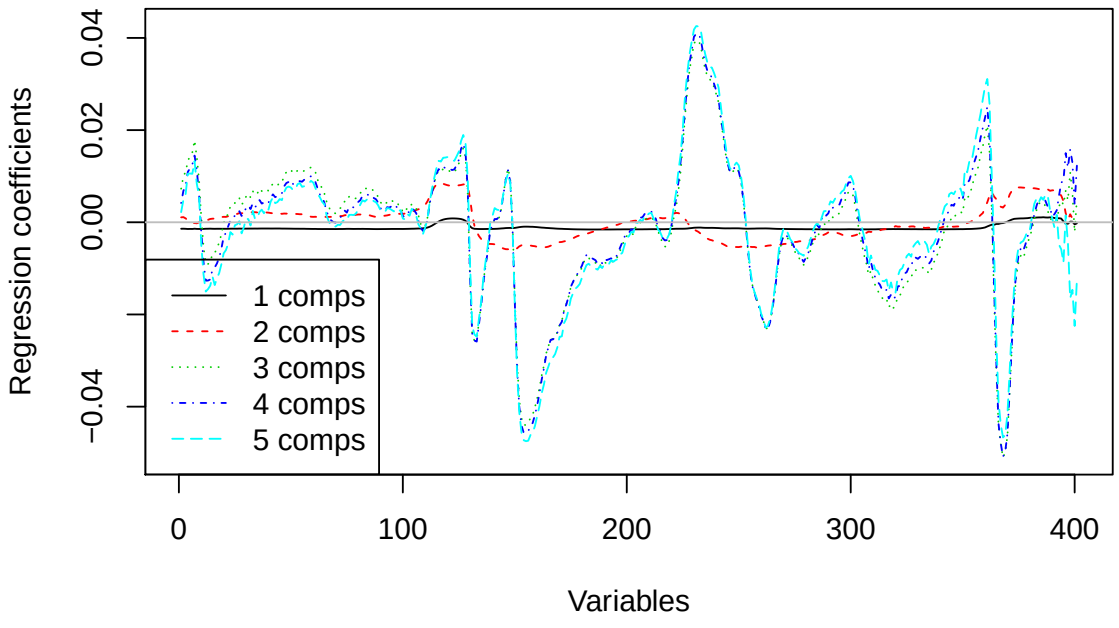
Graphically

```
plot(1:20, explvar(m.pcr), ylab='Percentage of explained variance',
       xlab='PCs', type='l', axes=FALSE)
axis(1, at=1:20)
axis(2)
```



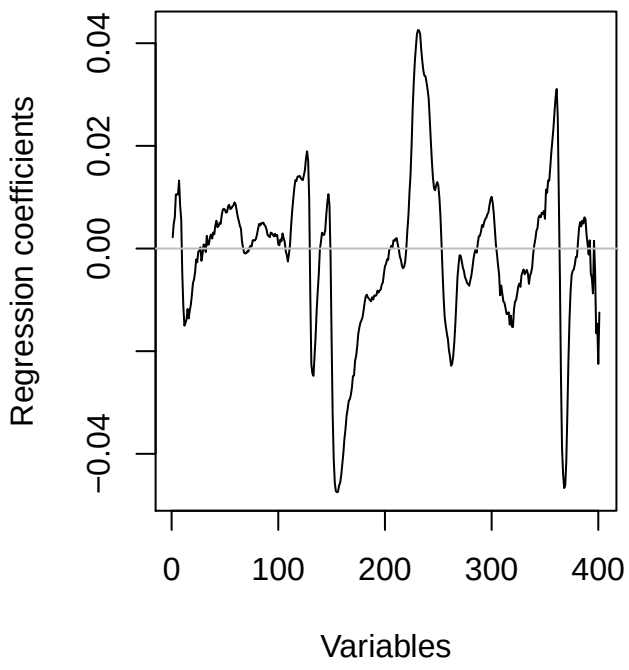
Plot the regression coefficients associated to the models with increasing PCs, from 1 to 5.

```
coefplot(m.pcr, ncomp=1:5, legendpos='bottomleft', main='',
        xlab='Variables', ylab='Regression coefficients')
```



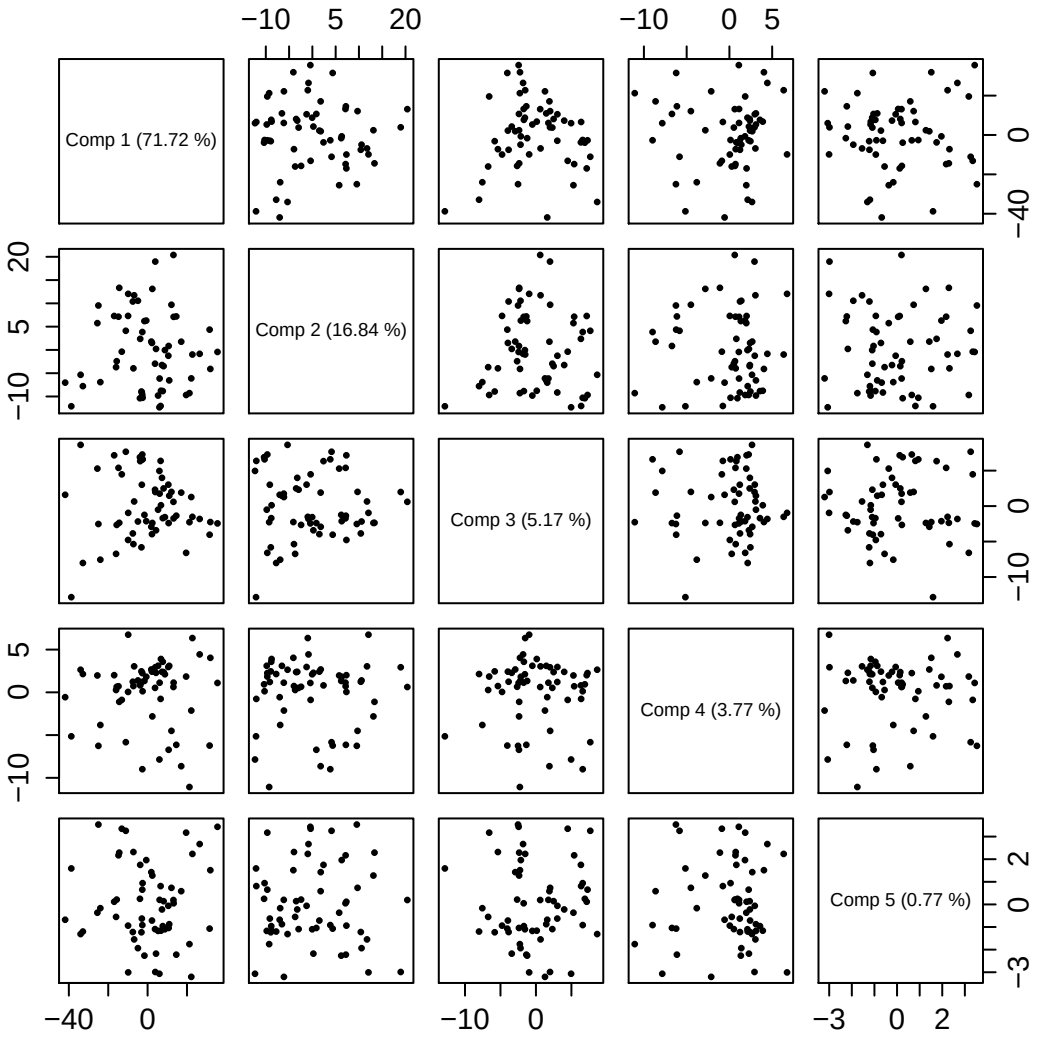
How can we comment?
Plot the regression coefficients for the model chosen by cross validation

```
coefplot(m.pcr, ncomp=5, main='', xlab='Variables', ylab='Regression coefficients')
```



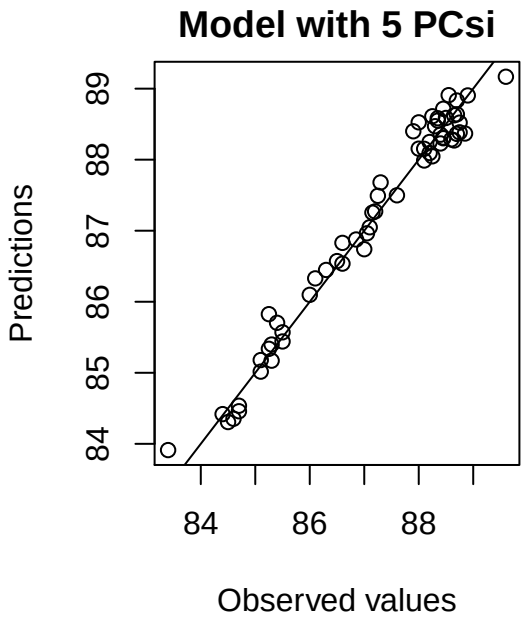
Evaluate the presence of groups of observations or outliers through the scores.

```
scoreplot(m.pcr, comps=1:5, cex=0.5, cex.lab=1.4, cex.axis=1.4, pch=19)
```



Increasing the number of PCs from 1 to 5 there are no anomalies of groups. The choice of 5 PCs seems satisfactory.
Finally, evaluate the predictions from the model

```
plot(m.pcr, xlab='Observed values', ylab='Predictions',
        main='Model with 5 PCsi')
abline(0, 1)
```



Values around the bisector suggest a good behavior of the model.
We can compare the results with those from ridge regression and lasso. Given the large amount of covariates using lasso could make more sense.

```
library(glmnet)
set.seed(222)
m.ridge <- glmnet(X, y, alpha=0, lambda.min=1e-4)
cv.ridge <- cv.glmnet(X, y, alpha=0, lambda.min=1e-4)
```

Option `lambda.min=1e-4` increases the grid of values used `R` for searching the optimum.

```
best.lambda <- cv.ridge$lambda.min
m.ridge.min <- glmnet(X, y, alpha=0, lambda=best.lambda)
min(cv.ridge$cvm)

## [1] 0.04543276
```

Basing on cross validation the MSE is 0.0454328, slightly lower than that from the previous model, that is equal to

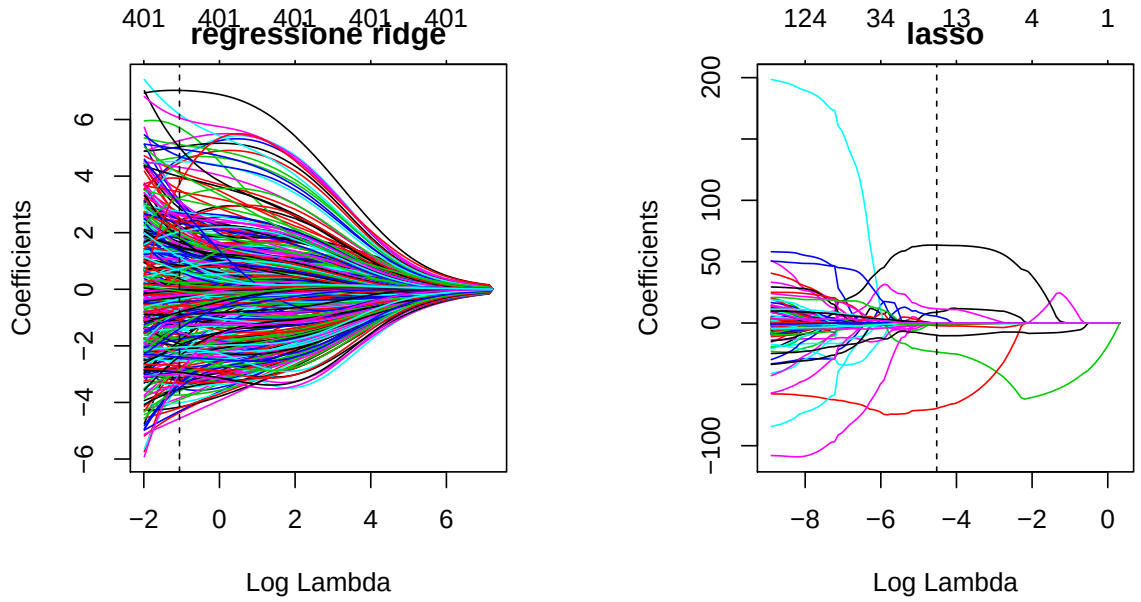
```
MSEP(m.pcr, ncomp=5)

##        (Intercept)  5 comps
## CV           2.381  0.05555
## adjCV        2.381  0.05465
```

The value is the square of the value obtained from function `summary(m.pcr)` (the function provides RMSEP).

```
m.lasso <- glmnet(X, y, alpha=1, lambda.min=1e-4)
set.seed(222)
cv.lasso <- cv.glmnet(X, y, alpha=1, lambda.min=1e-4)
best.lambda.lasso <- cv.lasso$lambda.min
m.lasso.min <- glmnet(X, y, alpha=1, lambda=best.lambda.lasso)
min(cv.lasso$cvm)

## [1] 0.04645222
```

The MSE from lasso is similar. Graphically compare ridge and lasso

```
par(mfrow=c(1,2))
plot(m.ridge, xvar='lambda', main='regressione ridge')
abline(v=log(best.lambda), lty=2)
plot(m.lasso, xvar='lambda', main='lasso')
abline(v=log(best.lambda.lasso), lty=2)
```



The reduction of coefficients from lasso is substantial:

```
id.zero <- which(coef(m.lasso.min)==0)
length(id.zero)

## [1] 381
```

There are 381 coefficients set to zero, so lasso selects 19 out of 401 covariates.