TD régression data vehicles

Julien JACQUES

12/15/2020

```
library(plsdepot)
data(vehicles)
```

Effectuons une régression linéaire:

```
modele=lm(price~.,data=vehicles)
summary(modele)
```

```
##
## Call:
## lm(formula = price ~ ., data = vehicles)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2919.8 -1410.1  -355.5  1418.8  5241.7
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -16321.524  42128.829  -0.387   0.7043
## diesel        1345.112   3988.023   0.337   0.7409
## turbo         1543.223   3167.584   0.487   0.6337
## two.doors      525.496   2182.582   0.241   0.8132
## hatchback    -7120.032   2830.591  -2.515   0.0247 *
## wheel.base      29.738    462.233   0.064   0.9496
## length        -371.484    254.303  -1.461   0.1662
## width          979.577    695.139   1.409   0.1806
## height        -494.126    695.891  -0.710   0.4893
## curb.weight     10.325     10.788   0.957   0.3548
## eng.size        83.775     96.394   0.869   0.3994
## horsepower      40.904     61.632   0.664   0.5177
## peak.rpm         2.265      1.954   1.159   0.2657
## symbol        1172.065    862.383   1.359   0.1956
## city.mpg       -89.821    545.087  -0.165   0.8715
## highway.mpg    128.438    464.388   0.277   0.7861
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2959 on 14 degrees of freedom
## Multiple R-squared:  0.9436, Adjusted R-squared:  0.8832
## F-statistic: 15.62 on 15 and 14 DF,  p-value: 3.19e-06
```

Le R^2 semble bon, proche de 1, alors que quasiment aucune variable n'est significative.

Examinons les VIFs :

```r
library(car)
```

```
## Loading required package: carData
```

```r
vif(modele)
```

```
##      diesel       turbo   two.doors   hatchback  wheel.base      length
##    8.721236    6.151519    3.791296    5.766553   34.465577   42.429644
##       width      height curb.weight    eng.size  horsepower    peak.rpm
##    9.529732    6.434802  126.458500   44.285608   19.798272    3.009453
##      symbol    city.mpg highway.mpg
##    3.160566   41.026076   31.572018
```

Il y a en effet plusieurs variables qui sont corrélées entre elles.

```r
library(MASS)
modele2=stepAIC(modele,direction = "both")
```

```
## Start:  AIC=488.68
## price ~ diesel + turbo + two.doors + hatchback + wheel.base +
##     length + width + height + curb.weight + eng.size + horsepower +
##     peak.rpm + symbol + city.mpg + highway.mpg
##
##                Df Sum of Sq       RSS    AIC
## - wheel.base    1     36232 122584300 486.69
## - city.mpg      1    237687 122785755 486.74
## - two.doors     1    507430 123055498 486.81
## - highway.mpg   1    669584 123217652 486.85
## - diesel        1    995819 123543886 486.93
## - turbo         1   2077686 124625754 487.19
## - horsepower    1   3855681 126403749 487.61
## - height        1   4413385 126961453 487.75
## - eng.size      1   6611702 129159770 488.26
## - curb.weight   1   8017751 130565819 488.59
## <none>                      122548068 488.68
## - peak.rpm      1  11764260 134312328 489.43
## - symbol        1  16168953 138717021 490.40
## - width         1  17382475 139930543 490.66
## - length        1  18678994 141227062 490.94
## - hatchback     1  55384497 177932565 497.87
##
## Step:  AIC=486.69
## price ~ diesel + turbo + two.doors + hatchback + length + width +
##     height + curb.weight + eng.size + horsepower + peak.rpm +
##     symbol + city.mpg + highway.mpg
##
##                Df Sum of Sq       RSS    AIC
## - city.mpg      1    201563 122785862 484.74
## - highway.mpg   1    634196 123218496 484.85
## - two.doors     1    660836 123245136 484.85
## - diesel        1   1193119 123777419 484.98
## - turbo         1   3167854 125752154 485.46
## - height        1   4517321 127101621 485.78
## - horsepower    1   4884503 127468803 485.87
## - eng.size      1   6667591 129251890 486.28
## <none>                      122584300 486.69
```

```
## - peak.rpm     1  12023419 134607719 487.50
## - curb.weight  1  12395279 134979579 487.58
## - symbol       1  16204153 138788453 488.42
## + wheel.base   1     36232 122548068 488.68
## - length       1  19206645 141790945 489.06
## - width        1  23686779 146271079 489.99
## - hatchback    1  57070536 179654836 496.16
##
## Step:  AIC=484.74
## price ~ diesel + turbo + two.doors + hatchback + length + width +
##     height + curb.weight + eng.size + horsepower + peak.rpm +
##     symbol + highway.mpg
##
##               Df Sum of Sq       RSS    AIC
## - two.doors    1    632951 123418813 482.90
## - highway.mpg  1    732227 123518090 482.92
## - diesel       1   1161817 123947680 483.03
## - turbo        1   2966389 125752252 483.46
## - height       1   4994171 127780033 483.94
## - eng.size     1   6466362 129252225 484.28
## - horsepower   1   8212543 130998405 484.68
## <none>                     122785862 484.74
## - curb.weight  1  12248331 135034193 485.60
## - peak.rpm     1  12637444 135423306 485.68
## + city.mpg     1    201563 122584300 486.69
## + wheel.base   1       107 122785755 486.74
## - symbol       1  17855651 140641513 486.82
## - length       1  19623273 142409135 487.19
## - width        1  23490540 146276402 487.99
## - hatchback    1  57006309 179792171 494.18
##
## Step:  AIC=482.9
## price ~ diesel + turbo + hatchback + length + width + height +
##     curb.weight + eng.size + horsepower + peak.rpm + symbol +
##     highway.mpg
##
##               Df Sum of Sq       RSS    AIC
## - highway.mpg  1    897154 124315968 481.11
## - diesel       1   1412288 124831101 481.24
## - turbo        1   3443014 126861827 481.72
## - height       1   4490560 127909374 481.97
## - eng.size     1   8010335 131429149 482.78
## <none>                     123418813 482.90
## - horsepower   1   9327567 132746381 483.08
## - curb.weight  1  11672361 135091175 483.61
## - peak.rpm     1  13316947 136735761 483.97
## + two.doors    1    632951 122785862 484.74
## + city.mpg     1    173677 123245136 484.85
## + wheel.base   1     53356 123365457 484.88
## - length       1  20638640 144057454 485.54
## - width        1  22972587 146391400 486.02
## - symbol       1  25505587 148924401 486.53
## - hatchback    1  62356082 185774895 493.17
##
```

```
## Step:  AIC=481.11
## price ~ diesel + turbo + hatchback + length + width + height +
##     curb.weight + eng.size + horsepower + peak.rpm + symbol
##
##               Df Sum of Sq       RSS    AIC
## - diesel       1   2023223 126339190 479.60
## - turbo        1   2988529 127304497 479.83
## - height       1   4546668 128862636 480.19
## - eng.size     1   8100400 132416368 481.01
## - horsepower   1   8450644 132766612 481.09
## <none>                     124315968 481.11
## - curb.weight  1  11014306 135330273 481.66
## - peak.rpm     1  12606383 136922351 482.01
## + highway.mpg  1    897154 123418813 482.90
## + two.doors    1    797878 123518090 482.92
## + city.mpg     1    416982 123898985 483.01
## + wheel.base   1    212357 124103611 483.06
## - length       1  22840705 147156673 484.17
## - width        1  26571829 150887796 484.93
## - symbol       1  26972474 151288441 485.01
## - hatchback    1  64239297 188555265 491.61
##
## Step:  AIC=479.6
## price ~ turbo + hatchback + length + width + height + curb.weight +
##     eng.size + horsepower + peak.rpm + symbol
##
##               Df Sum of Sq       RSS    AIC
## - height       1   3898236 130237427 478.51
## - turbo        1   5523729 131862919 478.88
## - horsepower   1   6463352 132802542 479.10
## - eng.size     1   7012015 133351205 479.22
## <none>                     126339190 479.60
## - peak.rpm     1  10591565 136930756 480.01
## + diesel       1   2023223 124315968 481.11
## + highway.mpg  1   1508089 124831101 481.24
## + two.doors    1   1207507 125131684 481.31
## + city.mpg     1    857154 125482036 481.39
## + wheel.base   1     60620 126278570 481.58
## - curb.weight  1  21436497 147775687 482.30
## - width        1  35372225 161711416 485.00
## - symbol       1  43985240 170324430 486.56
## - length       1  49702694 176041884 487.55
## - hatchback    1 117434156 243773346 497.32
##
## Step:  AIC=478.51
## price ~ turbo + hatchback + length + width + curb.weight + eng.size +
##     horsepower + peak.rpm + symbol
##
##               Df Sum of Sq       RSS    AIC
## - horsepower   1   5222511 135459938 477.69
## - turbo        1   6935969 137173396 478.07
## <none>                     130237427 478.51
## - peak.rpm     1  12784771 143022198 479.32
## + height       1   3898236 126339190 479.60
```

```
## + highway.mpg  1   1447992 128789435 480.17
## + diesel       1   1374791 128862636 480.19
## - curb.weight  1  17658007 147895434 480.32
## + city.mpg     1    482395 129755032 480.40
## - eng.size     1  18143853 148381280 480.42
## + wheel.base   1    163775 130073652 480.47
## + two.doors    1      1918 130235508 480.51
## - width        1  35299696 165537122 483.71
## - symbol       1  40656541 170893968 484.66
## - length       1  56729208 186966635 487.36
## - hatchback    1 117588100 247825527 495.81
##
## Step:  AIC=477.69
## price ~ turbo + hatchback + length + width + curb.weight + eng.size +
##     peak.rpm + symbol
##
##               Df Sum of Sq       RSS    AIC
## <none>                     135459938 477.69
## + horsepower   1   5222511 130237427 478.51
## - curb.weight  1  14246825 149706763 478.69
## + height       1   2657396 132802542 479.10
## + wheel.base   1   2594735 132865203 479.11
## + city.mpg     1    662921 134797017 479.54
## + two.doors    1     81936 135378002 479.67
## + diesel       1     53426 135406511 479.68
## + highway.mpg  1     18454 135441484 479.69
## - turbo        1  20555961 156015899 479.93
## - width        1  32129527 167589465 482.07
## - peak.rpm     1  38526714 173986652 483.20
## - symbol       1  48434625 183894563 484.86
## - eng.size     1  49967934 185427871 485.11
## - length       1  51736255 187196193 485.39
## - hatchback    1 112522581 247982519 493.83
```

Examinons le modèle obtenu :

```
summary(modele2)
```

```
##
## Call:
## lm(formula = price ~ turbo + hatchback + length + width + curb.weight +
##     eng.size + peak.rpm + symbol, data = vehicles)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3989.6 -1391.4  -480.5  1219.1  5306.4
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -35109.053  23964.875  -1.465 0.157727
## turbo         3140.737   1759.375   1.785 0.088691 .
## hatchback    -6760.217   1618.589  -4.177 0.000426 ***
## length         -433.600    153.104  -2.832 0.009984 **
## width          1049.308    470.161   2.232 0.036654 *
## curb.weight       8.848      5.953   1.486 0.152099
```

```
## eng.size       145.526     52.287   2.783 0.011140 *
## peak.rpm         2.848      1.166   2.444 0.023444 *
## symbol        1503.603    548.720   2.740 0.012264 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2540 on 21 degrees of freedom
## Multiple R-squared:  0.9377, Adjusted R-squared:  0.914
## F-statistic:  39.5 on 8 and 21 DF,  p-value: 5.974e-11
```

La variable curb.weight n'est pas significative, je l'enlève

```
modele3=lm(price ~ turbo + hatchback + length + width + eng.size + peak.rpm + symbol, data = vehicles)
summary(modele3)
```

```
##
## Call:
## lm(formula = price ~ turbo + hatchback + length + width + eng.size +
##     peak.rpm + symbol, data = vehicles)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3282.5 -1270.8  -730.7  1712.3  5787.4
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -54010.276  20862.105  -2.589 0.016751 *
## turbo         4648.590   1476.308   3.149 0.004661 **
## hatchback    -5686.831   1487.765  -3.822 0.000929 ***
## length        -263.948    104.792  -2.519 0.019548 *
## width         1086.271    482.227   2.253 0.034593 *
## eng.size       217.736     19.838  10.976 2.16e-10 ***
## peak.rpm         2.797      1.197   2.337 0.028928 *
## symbol        1247.061    534.977   2.331 0.029314 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2609 on 22 degrees of freedom
## Multiple R-squared:  0.9311, Adjusted R-squared:  0.9092
## F-statistic:  42.5 on 7 and 22 DF,  p-value: 2.442e-11
```

Tout est significatif. On peut comparrer les critères AIC (attention, avec la même fonction, par exemple AIC, car suivant les fonctions de R certains variantes du critères AIC peuvent être implémentée.)

```
AIC(modele2)
```

```
## [1] 564.826
```

```
AIC(modele3)
```

```
## [1] 565.8261
```

Même si le AIC est légérement moins bon pour le modèle3, toutes les variables sont significatives donc je le préférerai.

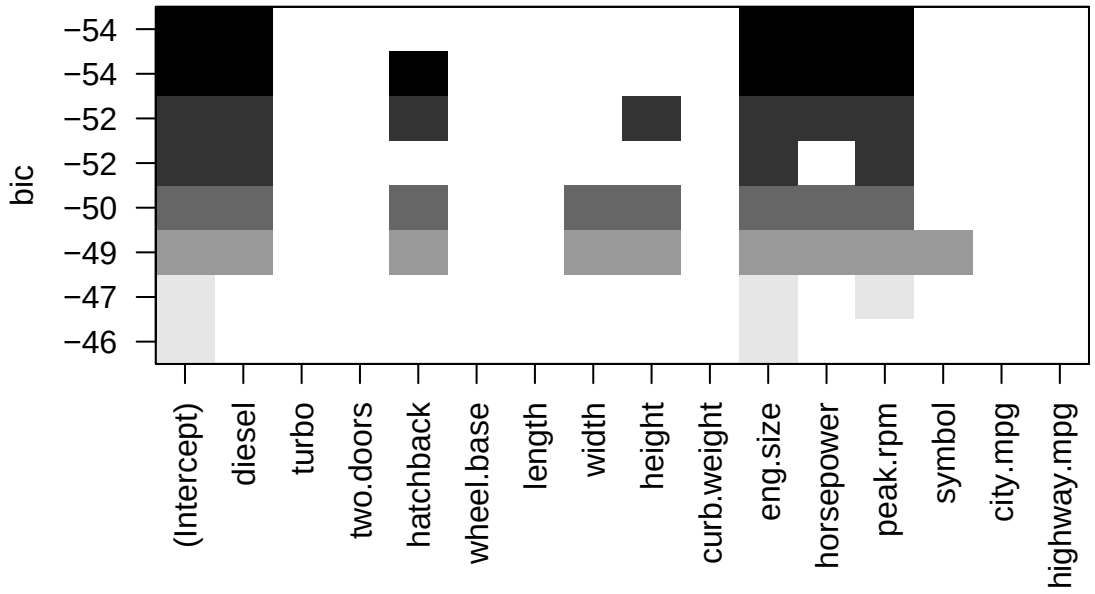
Examinons pour finir les VIF, ils sont corrects.

```
vif(modele3)
```

```
##     turbo hatchback    length     width  eng.size  peak.rpm    symbol
##  1.718858  2.049228  9.267903  5.899295  2.412832  1.452460  1.564567
```

On peut aussi utiliser la librairie suivante, qui permet notamment de tracer pour les meilleurs modèles parcourus l'importance de chaque variable :

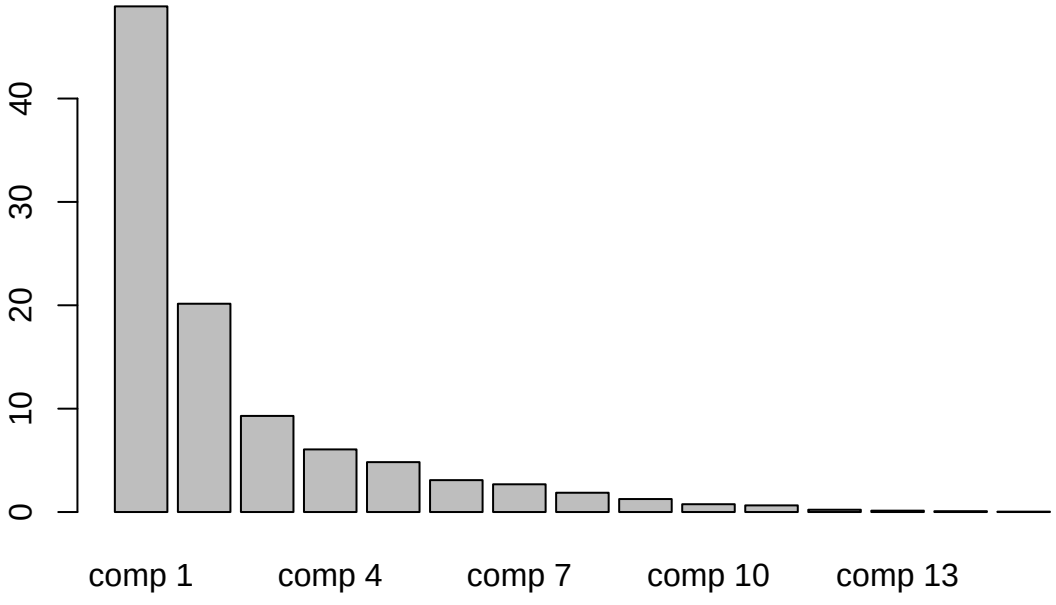
```
library(leaps)
modele4bis=regsubsets(price ~ .,method="forward", data = vehicles)
plot(modele4bis)
```



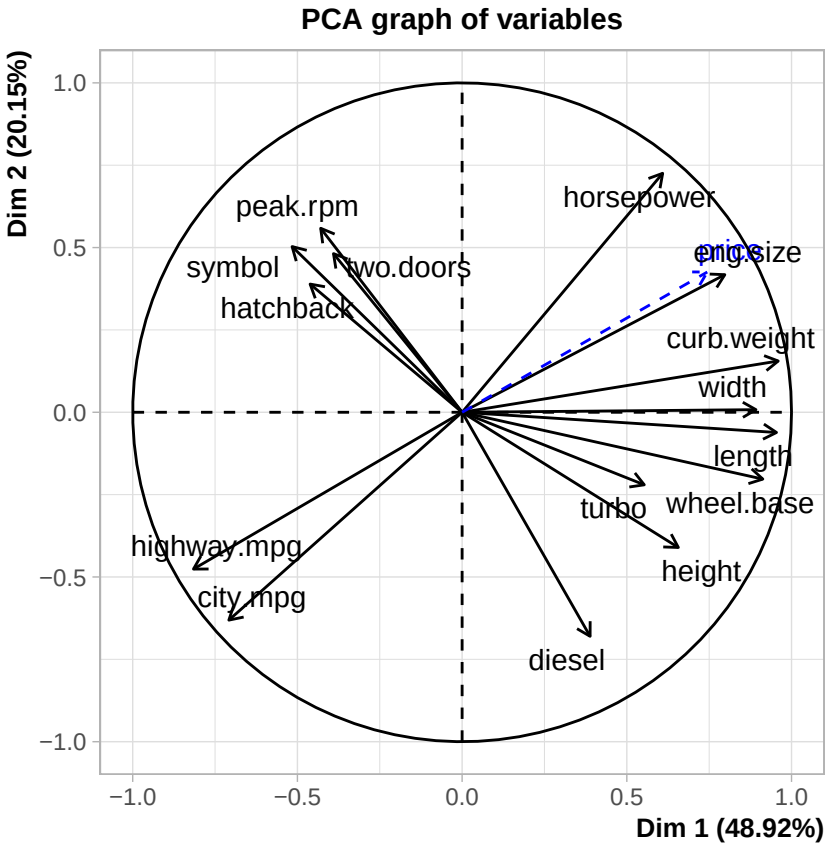
Regression sur composantes principales

On peut commencer par effectuer une ACP :

```
library(FactoMineR)
acp=PCA(vehicles,quanti.sup = 13,graph = FALSE)
barplot(acp$eig[,2])
```



```
plot(acp,choix="var")
```



Des stratégies classiques de sélection du nombre d'axe en ACP nous conduirait à choisir 3 axes. On peut aussi sélectionner ce nombre en fonction de la qualité du modèle de régression.

La régression sur composante principale peut-être réalisée directement

```
library(pls)
```

```
## Warning: package 'pls' was built under R version 4.4.1
```

```
##
## Attaching package: 'pls'
```

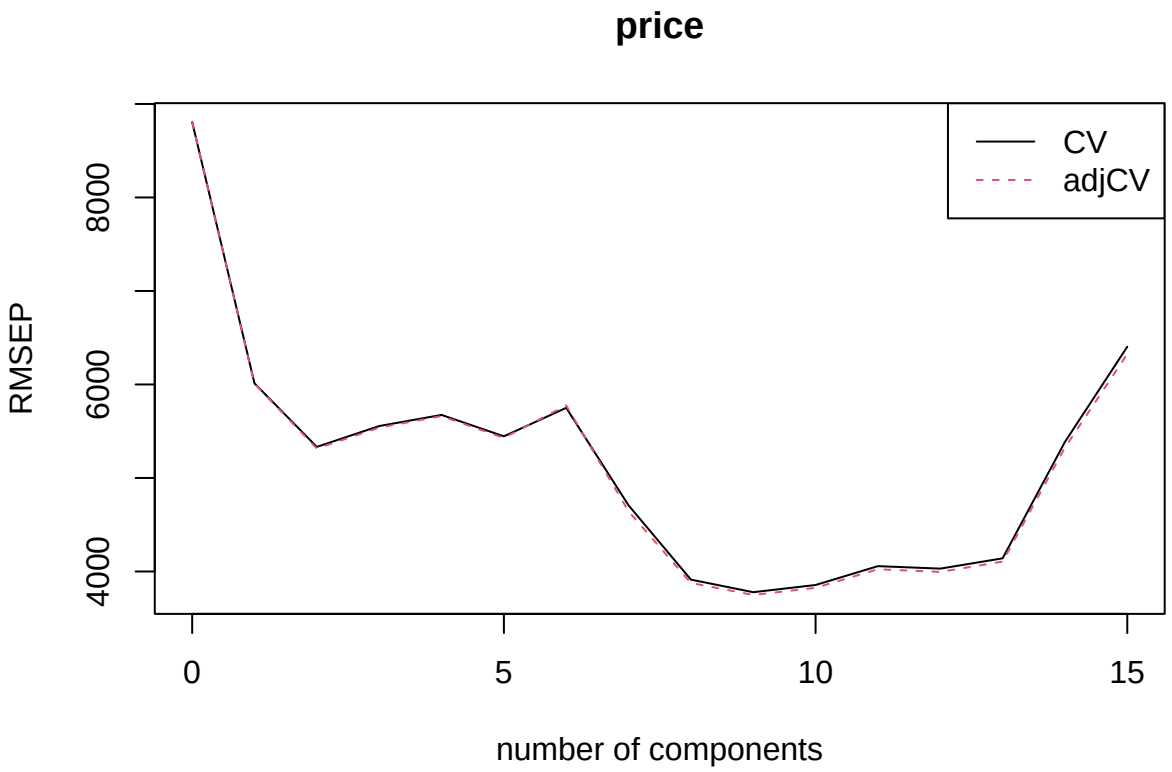
```
## The following object is masked from 'package:stats':
##
##     loadings
```

```
modele5=pcr(price~.,data=vehicles,validation='LOO',scale=TRUE)
summary(modele5)
```

```
## Data:    X dimension: 30 15
##  Y dimension: 30 1
## Fit method: svdpc
## Number of components considered: 15
##
## VALIDATION: RMSEP
## Cross-validated using 30 leave-one-out segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            8806     6012     5332     5556     5674     5445     5750
```

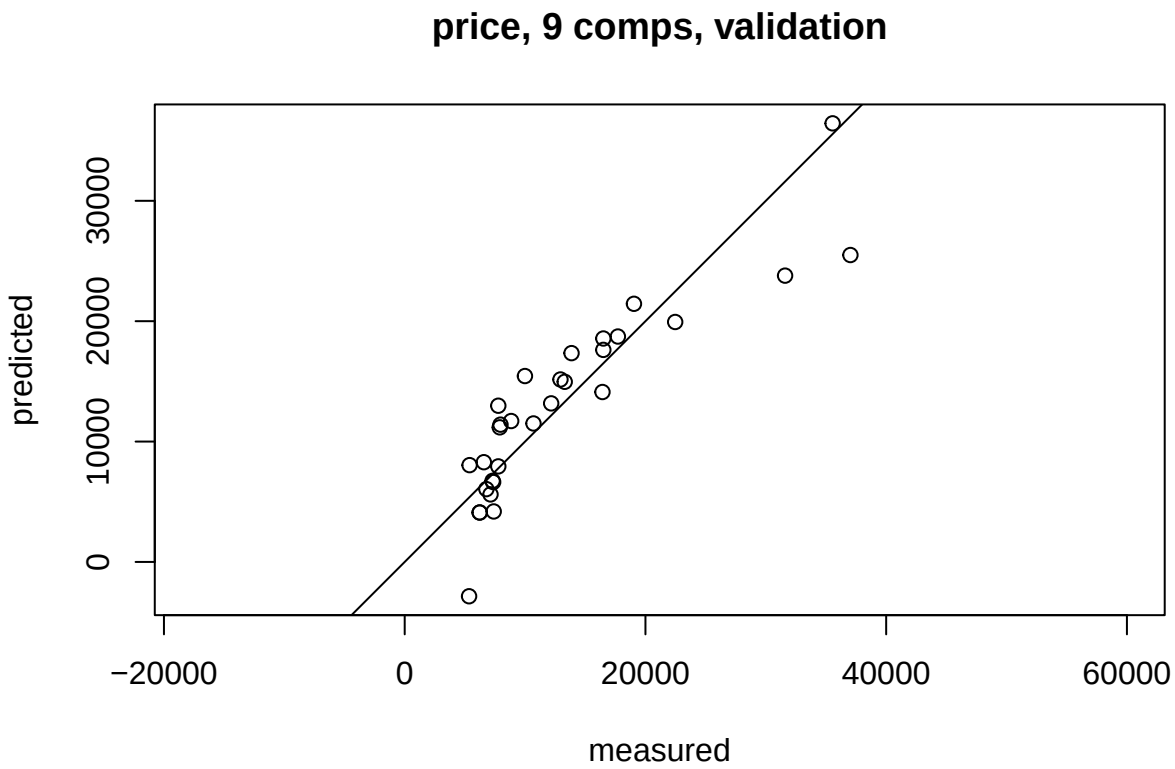
```
## adjCV         8806     6007     5315     5538     5662     5429     5775
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        4707     3913     3778      3855      4057      4030      4141
## adjCV     4643     3877     3749      3825      4023      3995      4105
##        14 comps  15 comps
## CV         5386      6403
## adjCV      5325      6326
##
## TRAINING: % variance explained
##         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X         48.92    69.06    78.36    84.42    89.25    92.33    95.01    96.88
## price     55.39    73.52    73.56    75.97    78.87    82.64    92.29    92.85
##         9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X         98.13     98.89     99.53     99.75     99.88     99.96    100.00
## price     93.21     93.22     93.28     93.93     93.93     94.19     94.36
```

```
plot(RMSEP(modele5), legendpos = "topright")
```



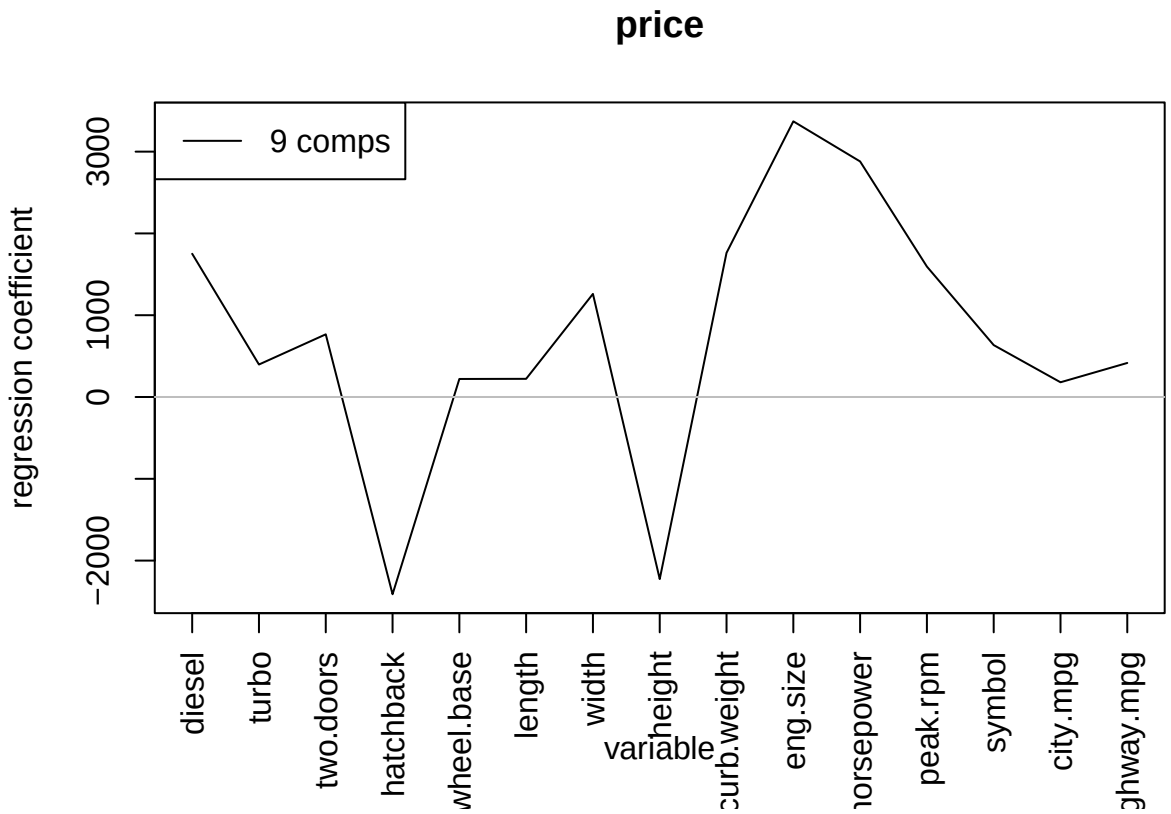
Le bon nombre de composantes semble être 9. On peut regarder la qualité de la prédiction de ce modèle en traçant les valeurs prédites en fonctions des valeurs mesurées :

```
plot(modele5, ncomp = 9, asp = 1, line = TRUE)
```



L'examen des coefficients de régression des composantes principales ne nous sera d'aucune utilité. Par contre, ces composantes étant elles-mêmes des combinaisons linéaires des variables initiales, on peut reconstruire les coefficients de régressions sur les variables initiales :

```
plot(modele5, plottype = "coef", ncomp=9, legendpos = "topleft",xaxt='n')
axis(1,at=1:15,labels=colnames(vehicles)[-13],las = 2)
```



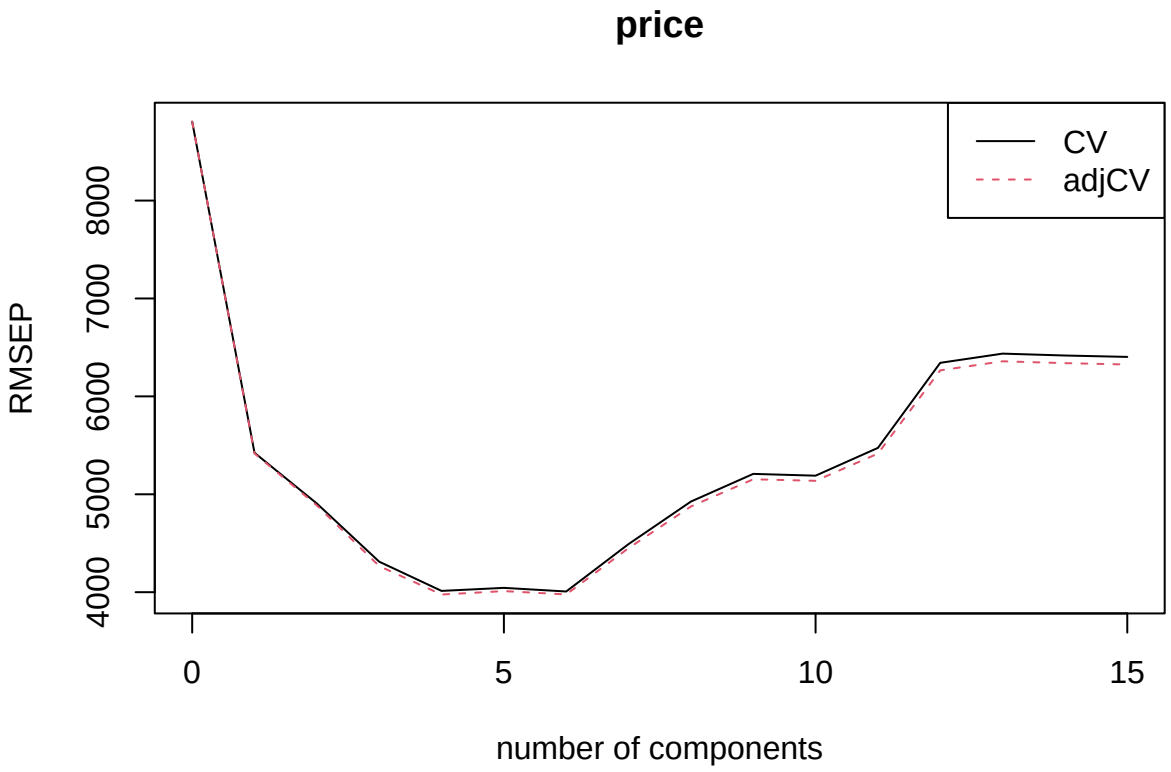
Regression sur composantes PLS

```
library(pls)
modele6=plsr(price~.,data=vehicles,validation='LOO',scale=TRUE)
summary(modele6)
```

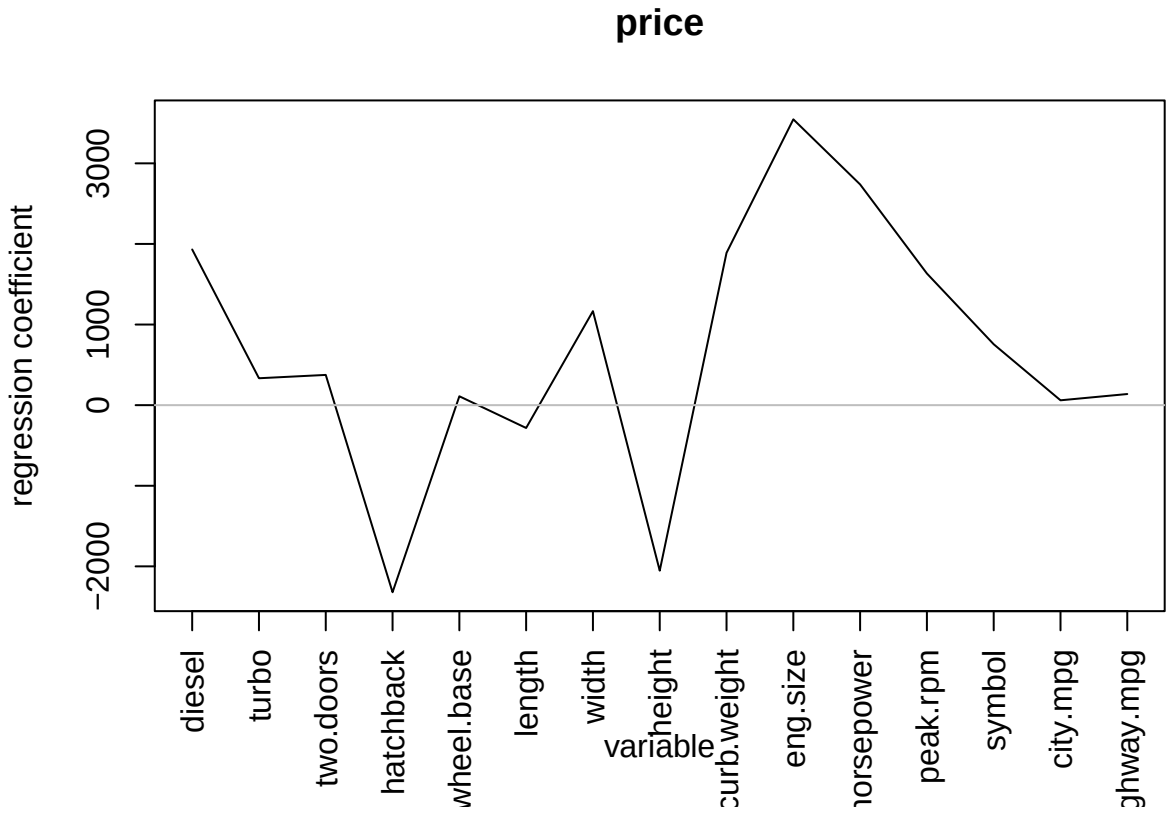
```
## Data:    X dimension: 30 15
##  Y dimension: 30 1
## Fit method: kernelpls
## Number of components considered: 15
##
## VALIDATION: RMSEP
## Cross-validated using 30 leave-one-out segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            8806     5424     4906     4312     4013     4044     4007
## adjCV         8806     5417     4886     4267     3976     4011     3977
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        4491     4926     5208      5190      5474      6343      6437
## adjCV     4450     4876     5153      5137      5416      6266      6358
##        14 comps  15 comps
## CV         6417      6403
## adjCV      6339      6326
##
## TRAINING: % variance explained
##        1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X        47.32    67.78    73.14    77.87    84.81    89.60    92.93    94.71
## price    70.47    82.18    92.06    93.32    93.46    93.64    93.84    94.07
##        9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
```

```
## X         96.58     98.07     98.84     99.09     99.78     99.87    100.00
## price     94.17     94.21     94.26     94.34     94.35     94.36     94.36
```

```r
plot(RMSEP(modele6), legendpos = "topright")
```




```r
plot(modele6, plottype = "coef", ncomp=4,xaxt='n')
axis(1,at=1:15,labels=colnames(vehicles)[-13],las = 2)
```


price
regression coefficient
3000
1000
0
−2000
diesel
turbo
two.doors
hatchback
wheel.base
length
width
height
curb.weight
eng.size
horsepower
peak.rpm
symbol
city.mpg
ghway.mpg
variable