

Régression linéaire multiple

--

Détection des points atypiques

et

Sélection de variables

Ricco.Rakotomalala

<http://eric.univ-lyon2.fr/~ricco/cours>

Fichier de données

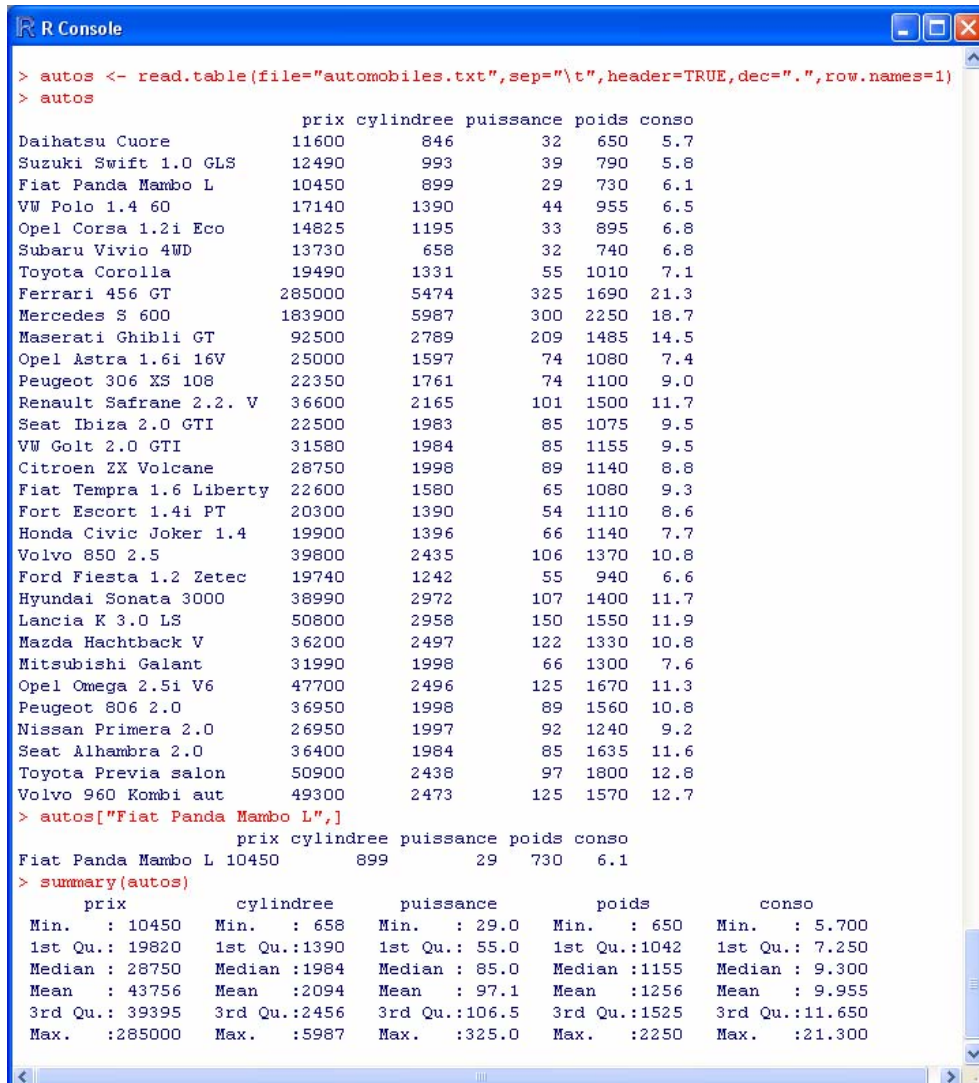
Prédiction de la consommation de véhicules

- (1) Prédire la consommation des véhicules à partir de ses caractéristiques
- (2) Diagnostic de la régression avec les graphiques des résidus
- (3) Détection et traitement des points atypiques
- (4) Détection de la colinéarité
- (5) Sélection de variables

| modele | prix | cylindree | puissance | Poids | conso |
|-------------------------|--------|-----------|-----------|-------|-------|
| Daihatsu Cuore | 11600 | 846 | 32 | 650 | 5.7 |
| Suzuki Swift 1.0 GLS | 12490 | 993 | 39 | 790 | 5.8 |
| Fiat Panda Mambo L | 10450 | 899 | 29 | 730 | 6.1 |
| VW Polo 1.4 60 | 17140 | 1390 | 44 | 955 | 6.5 |
| Opel Corsa 1.2i Eco | 14825 | 1195 | 33 | 895 | 6.8 |
| Subaru Vivio 4WD | 13730 | 658 | 32 | 740 | 6.8 |
| Toyota Corolla | 19490 | 1331 | 55 | 1010 | 7.1 |
| Ferrari 456 GT | 285000 | 5474 | 325 | 1690 | 21.3 |
| Mercedes S 600 | 183900 | 5987 | 300 | 2250 | 18.7 |
| Maserati Ghibli GT | 92500 | 2789 | 209 | 1485 | 14.5 |
| Opel Astra 1.6i 16V | 25000 | 1597 | 74 | 1080 | 7.4 |
| Peugeot 306 XS 108 | 22350 | 1761 | 74 | 1100 | 9.0 |
| Renault Safrane 2.2. V | 36600 | 2165 | 101 | 1500 | 11.7 |
| Seat Ibiza 2.0 GTI | 22500 | 1983 | 85 | 1075 | 9.5 |
| VW Golt 2.0 GTI | 31580 | 1984 | 85 | 1155 | 9.5 |
| Citroen ZX Volcane | 28750 | 1998 | 89 | 1140 | 8.8 |
| Fiat Tempra 1.6 Liberty | 22600 | 1580 | 65 | 1080 | 9.3 |
| Fort Escort 1.4i PT | 20300 | 1390 | 54 | 1110 | 8.6 |
| Honda Civic Joker 1.4 | 19900 | 1396 | 66 | 1140 | 7.7 |
| Volvo 850 2.5 | 39800 | 2435 | 106 | 1370 | 10.8 |
| Ford Fiesta 1.2 Zetec | 19740 | 1242 | 55 | 940 | 6.6 |
| Hyundai Sonata 3000 | 38990 | 2972 | 107 | 1400 | 11.7 |
| Lancia K 3.0 LS | 50800 | 2958 | 150 | 1550 | 11.9 |
| Mazda Hachtback V | 36200 | 2497 | 122 | 1330 | 10.8 |
| Mitsubishi Galant | 31990 | 1998 | 66 | 1300 | 7.6 |
| Opel Omega 2.5i V6 | 47700 | 2496 | 125 | 1670 | 11.3 |
| Peugeot 806 2.0 | 36950 | 1998 | 89 | 1560 | 10.8 |
| Nissan Primera 2.0 | 26950 | 1997 | 92 | 1240 | 9.2 |
| Seat Alhambra 2.0 | 36400 | 1984 | 85 | 1635 | 11.6 |
| Toyota Previa salon | 50900 | 2438 | 97 | 1800 | 12.8 |
| Volvo 960 Kombi aut | 49300 | 2473 | 125 | 1570 | 12.7 |

Chargement des données et statistiques descriptives

```
#dans ce format, le séparateur est tabulation, la première ligne contient
#le nom des variables, le point décimal est ".", la première colonne est
#le nom des observations
autos <- read.table(file="automobiles.txt",sep="\t",header=TRUE,dec=".",row.names=1)
#pour afficher les données
autos
#pour afficher les valeurs de la Fiat Panda Mambo L, on peut écrire
autos["Fiat Panda Mambo L",]
#pour obtenir la liste des noms de véhicules
row.names(autos)
#petit résumé des données
summary(autos)
#nombre de lignes et de colonnes dans le data.frame
print(nrow(autos))
print(ncol(autos))
```

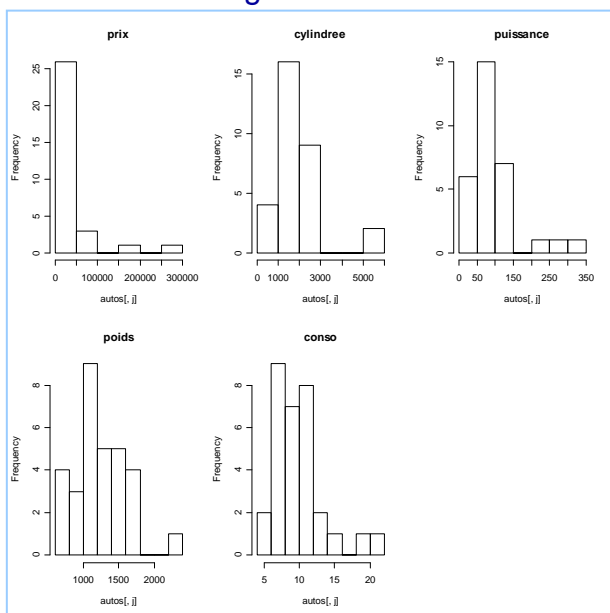


```
R Console
> autos <- read.table(file="automobiles.txt",sep="\t",header=TRUE,dec=".",row.names=1)
> autos
      prix cylindree puissance poids conso
Daihatsu Cuore      11600      846      32    650  5.7
Suzuki Swift 1.0 GLS 12490      993      39    790  5.8
Fiat Panda Mambo L  10450      899      29    730  6.1
VW Polo 1.4 60     17140     1390      44    955  6.5
Opel Corsa 1.2i Eco 14825     1195      33    895  6.8
Subaru Vivio 4WD   13730      658      32    740  6.8
Toyota Corolla     19490     1331      55   1010  7.1
Ferrari 456 GT     285000    5474     325   1690 21.3
Mercedes S 600    183900    5987     300  2250 18.7
Maserati Ghibli GT  92500     2789     209   1485 14.5
Opel Astra 1.6i 16V 25000     1597      74   1080  7.4
Peugeot 306 XS 108 22350     1761      74   1100  9.0
Renault Safrane 2.2. V 36600     2165     101   1500 11.7
Seat Ibiza 2.0 GTI  22500     1983      85   1075  9.5
VW Golt 2.0 GTI    31580     1984      85   1155  9.5
Citroen ZX Volcane 28750     1998      89   1140  8.8
Fiat Temptra 1.6 Liberty 22600     1580      65   1080  9.3
Ford Escort 1.4i PT 20300     1390      54   1110  8.6
Honda Civic Joker 1.4 19900     1396      66   1140  7.7
Volvo 850 2.5     39800     2435     106   1370 10.8
Ford Fiesta 1.2 Zetec 19740     1242      55    940  6.6
Hyundai Sonata 3000 38990     2972     107   1400 11.7
Lancia K 3.0 LS   50800     2958     150   1550 11.9
Mazda Hachtback V  36200     2497     122   1330 10.8
Mitsubishi Galant  31990     1998      66   1300  7.6
Opel Omega 2.5i V6 47700     2496     125   1670 11.3
Peugeot 806 2.0   36950     1998      89   1560 10.8
Nissan Primera 2.0 26950     1997      92   1240  9.2
Seat Alhambra 2.0  36400     1984      85   1635 11.6
Toyota Previa salon 50900     2438      97   1800 12.8
Volvo 960 Kombi aut 49300     2473     125   1570 12.7
> autos["Fiat Panda Mambo L",]
      prix cylindree puissance poids conso
Fiat Panda Mambo L 10450      899      29    730  6.1
> summary(autos)
      prix      cylindree      puissance      poids      conso
Min.   : 10450   Min.   : 658     Min.   : 29.0   Min.   : 650   Min.   : 5.700
1st Qu.: 19820   1st Qu.:1390     1st Qu.: 55.0   1st Qu.:1042   1st Qu.: 7.250
Median : 28750   Median :1984     Median : 85.0   Median :1155   Median : 9.300
Mean   : 43756   Mean  :2094     Mean  : 97.1   Mean  :1256   Mean  : 9.955
3rd Qu.: 39395   3rd Qu.:2456     3rd Qu.:106.5   3rd Qu.:1525   3rd Qu.:11.650
Max.   :285000   Max.   :5987     Max.   :325.0   Max.   :2250   Max.   :21.300
```

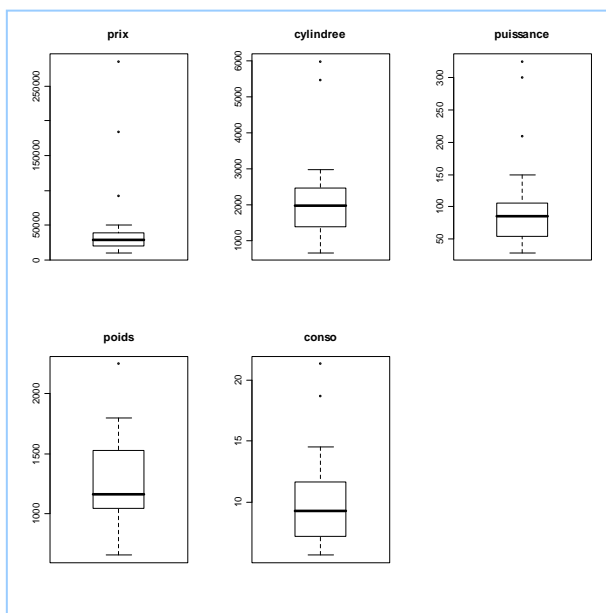
Statistiques descriptives - Graphiques

```
#histogramme des variables
#fractionne la fenêtre graphique en portions rectangulaires
par(mfrow=c(2,3))
for (j in 1:5) {hist(autos[,j],main=names(autos)[j])}
#boxplot des variables
par(mfrow=c(2,3))
for (j in 1:5) {boxplot(autos[,j],main=names(autos)[j])}
#supprimer le fractionnement
layout(1)
#nuage de points 2 à 2
pairs(autos)
```

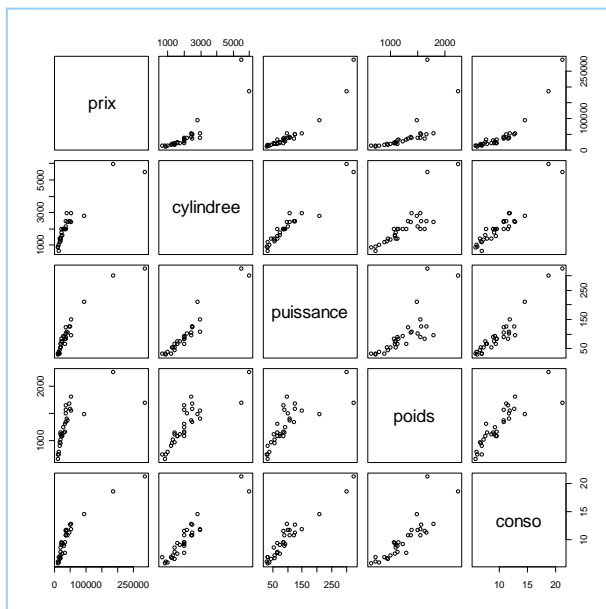
Histogrammes



Boîtes à moustaches



Nuages de points (2 à 2)



Régression linéaire multiple

```
reg <- lm(conso ~ prix + cylindree + puissance + poids, autos)
print(reg)
attributes(reg)
resume <- summary(reg)
print(resume)
attributes(resume)
print(resume$coefficients)
nrow(resume$coefficients)
print(resume$coefficients[1,"Std. Error"])
print(resume$coefficients[1,2])
```

Lancer la régression, on obtient un objet « régression » de type « list », on a accès aux champs que l'on peut manipuler directement

```
R Console
> reg <- lm(conso ~ prix + cylindree + puissance + poids, autos)
> print(reg)

Call:
lm(formula = conso ~ prix + cylindree + puissance + poids, data = autos)

Coefficients:
(Intercept)      prix      cylindree      puissance      poids
 2.456e+00    2.042e-05   -5.006e-04    2.499e-02    4.161e-03

> attributes(reg)
$names
[1] "coefficients" "residuals" "effects" "rank" "fitted.values" "assign"
[7] "qr" "df.residual" "xlevels" "call" "terms" "model"

$class
[1] "lm"

> resume <- summary(reg)
> print(resume)

Call:
lm(formula = conso ~ prix + cylindree + puissance + poids, data = autos)

Residuals:
    Min       1Q   Median       3Q      Max
-1.5678 -0.6704  0.1183  0.5283  1.4360

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.456e+00  6.268e-01  3.919 0.000578 ***
prix         2.042e-05  8.731e-06  2.339 0.027297 *
cylindree   -5.006e-04  5.748e-04 -0.871 0.391797
puissance    2.499e-02  9.992e-03  2.501 0.018993 *
poids        4.161e-03  8.788e-04  4.734 6.77e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8172 on 26 degrees of freedom
Multiple R-Squared:  0.9546,    Adjusted R-squared:  0.9476
F-statistic: 136.5 on 4 and 26 DF,  p-value: < 2.2e-16

> attributes(resume)
$names
[1] "call" "terms" "residuals" "coefficients" "aliased" "sigma"
[7] "df" "r.squared" "adj.r.squared" "fstatistic" "cov.unscaled"

$class
[1] "summary.lm"

> print(resume$coefficients)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.456294e+00 6.268181e-01  3.9186711 5.777629e-04
prix         2.042054e-05 8.730670e-06  2.3389432 2.729732e-02
cylindree   -5.005933e-04 5.748221e-04 -0.8708666 3.917972e-01
puissance    2.499448e-02 9.991852e-03  2.5014857 1.899350e-02
poids        4.160583e-03 8.787869e-04  4.7344612 6.774484e-05
> nrow(resume$coefficients)
[1] 5
> print(resume$coefficients[1,"Std. Error"])
[1] 0.6268181
> print(resume$coefficients[1,2])
[1] 0.6268181
> |
<
```

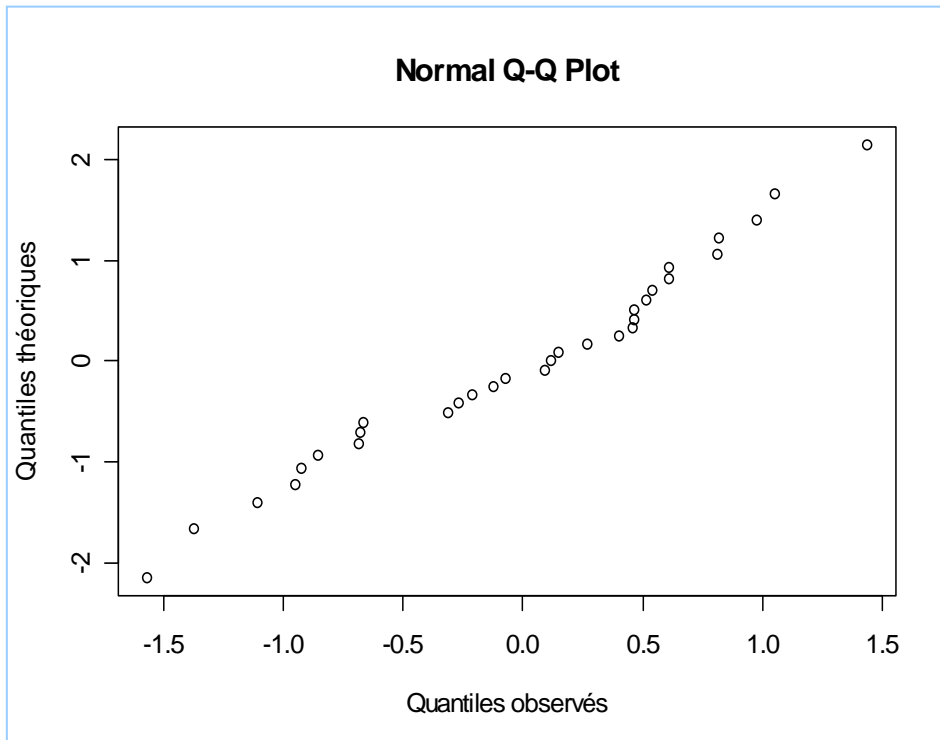
Objet « résumé de la régression ». On peut accéder à ses champs, que l'on peut aussi manipuler directement (ex. les coefficients de la régression, les écarts-type des coefficients estimés, le t de Student, le R², R² ajusté, etc.)

Exemple de manipulation directe des coefficients issus de la régression à partir de l'objet « résumé ». Ici on affiche l'écart-type de la constante, de deux manières différentes

Graphiques des résidus (normalité)

```
#récupérer les résidus (à partir des champs de l'objet reg)
e <- reg$residuals #ceci aurait été valable aussi : e <- residuals(reg)

#droite de Henry, attention R conserve les quantiles normalisés en ordonnée
qqnorm(e,dataax=TRUE,ylab="Quantiles observés",xlab="Quantiles théoriques")
```



Le graphique QQ-plot permet de vérifier la normalité d'une distribution.

Il prend comme point de départ la fonction de répartition observée et compare (en abscisse) les quantiles observés et (en ordonnée) les quantiles obtenus si la distribution suivait une loi normale. S'ils concordent (forment une droite), on peut dire que la distribution est compatible avec la loi normale.

Nous nous en servons pour vérifier si l'hypothèse de normalité des résidus à la base de tout le dispositif inférentiel de la régression est crédible sur nos données : il semble que OUI.

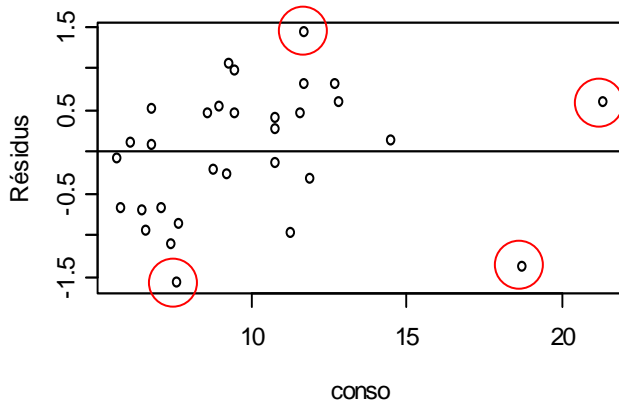
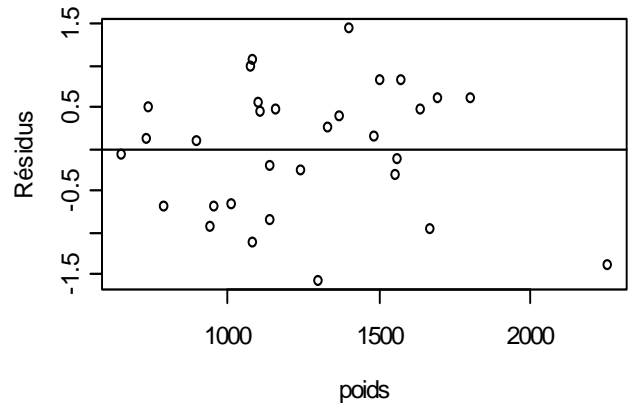
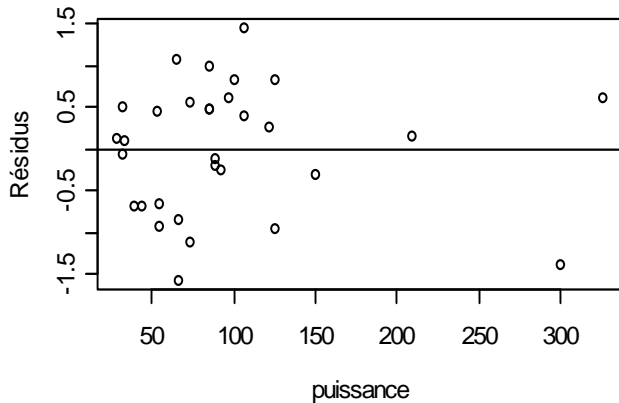
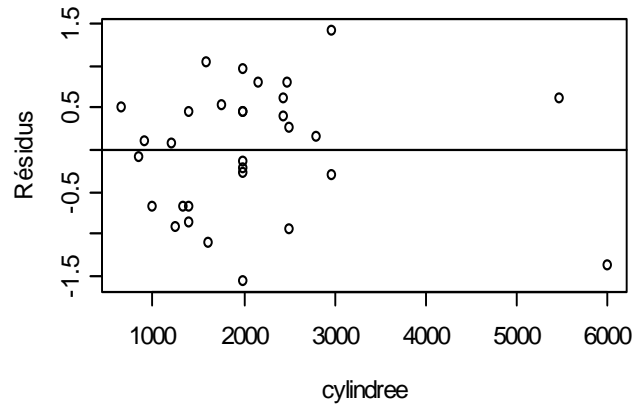
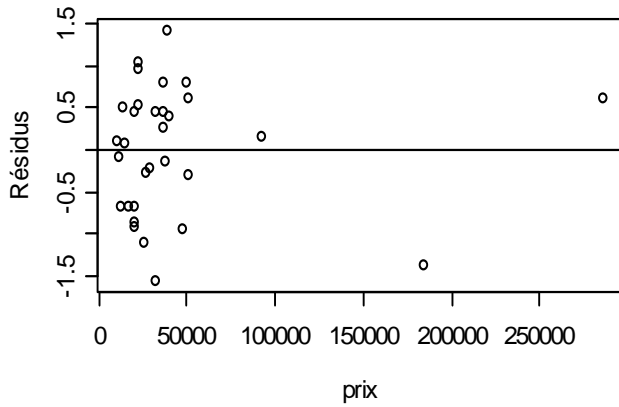
Graphiques des résidus (résidus vs. Variables)

```
#graphiques des résidus
```

```
par(mfrow=c(3,2))
```

```
for (j in 1:5){plot(autos[,j],e,ylab="Résidus",xlab=names(autos)[j]); abline(h=0)}
```

```
layout(1)#réinitialiser après coup l'espace graphique
```



Les graphiques des résidus (en ordonnée) vs. les variables de l'étude (en abscisse) permet de détecter visuellement les points atypiques :

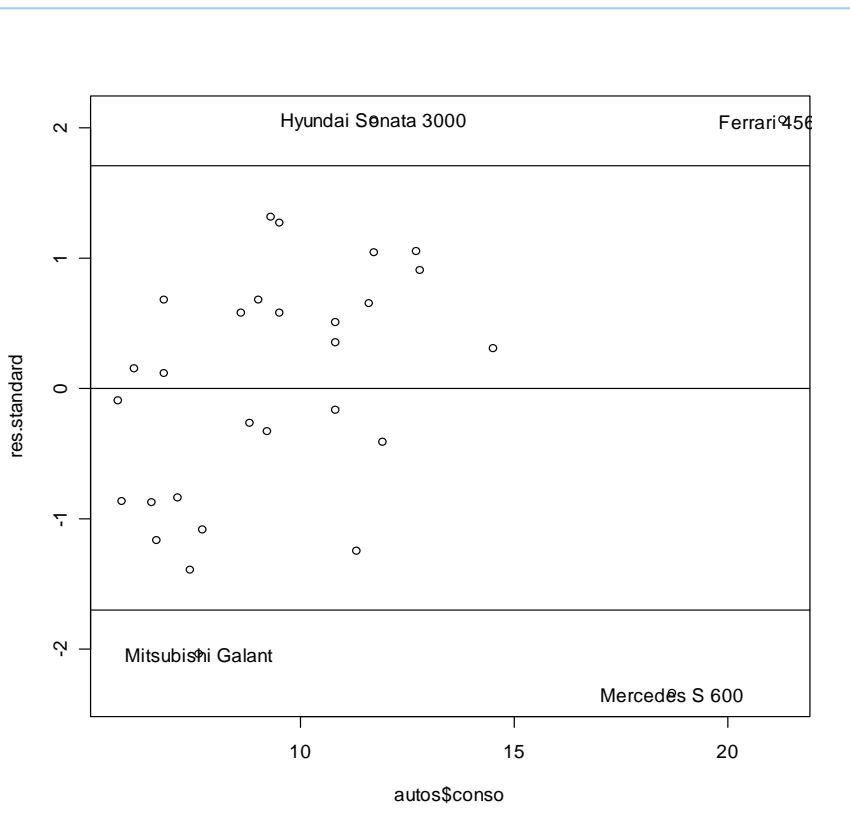
- (1) tant sur les variables (les points à la périphérie en abscisse) que
- (2) dans la régression (les points à la périphérie en ordonnée).

Ex. pour la variable endogène CONSO, que se passe-t-il pour ces points ?

Étude des points atypiques

Le résidu standardisé

```
#calcul du résidu standardisé
res.standard <- rstandard(reg)
#risque alpha = 0.1
alpha <- 0.1
#calcul du seuil à partir de la loi de Student à (n-p-1) ddl
seuil.standard <- qt(1-alpha/2,31-4-1)
#construction du graphique des résidus standardisés
plot(autos$conso,res.standard)
#ajout des seuils dans le graphique
abline(h=-seuil.standard)
abline(h+=seuil.standard)
abline(h=0)
#détection des points en dehors des tuyaux, on obtient le tableau des obs. atypiques
ab.standard <- autos[res.standard < -seuil.standard | res.standard > +seuil.standard,]
#mettre en évidence les points atypiques dans le graphique
for (i in 1:nrow(ab.standard)){
  #on récupère une chaîne de carac., la désignation du véhicule
  vehicule <- row.names(ab.standard)[i]
  #on place le point atypique (au sens du résidu standardisé) dans le graphique
  text(autos[vehicule,"conso"],res.standard[vehicule],vehicule)
}
```



Construction du graphique « endogène vs. Résidus standardisé »

Ajout dans ce graphique des limites permettant de statuer sur le caractère atypique d'un résidu

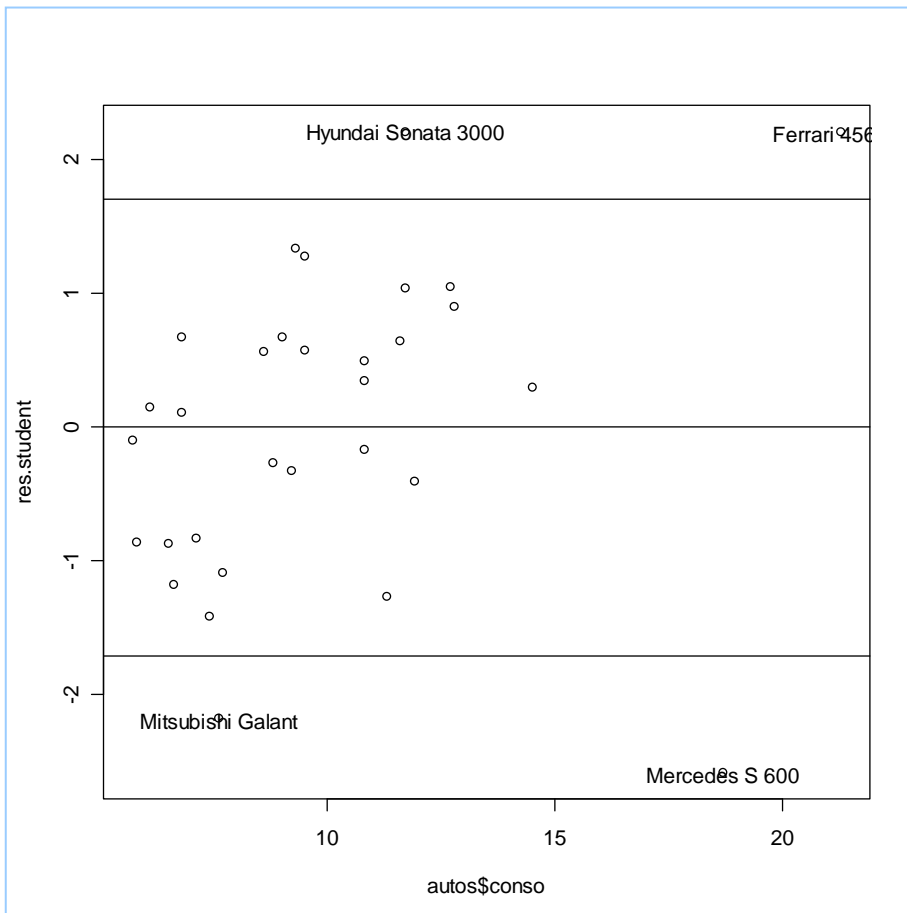
Détection automatique des observations atypiques au sens du résidu standardisé

Ajout de la désignation de ces observations dans le graphique

Étude des points atypiques

Le résidu studentisé (même démarche mais indicateur et d.d.l différents)

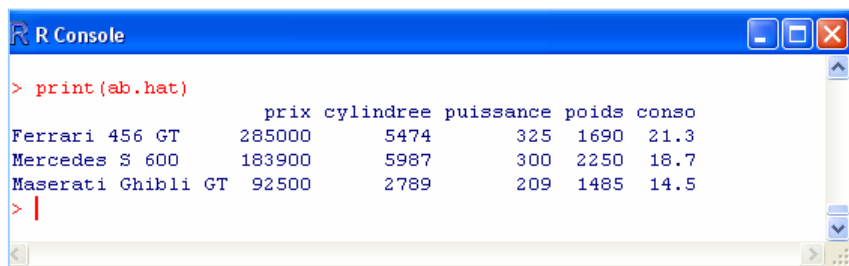
```
#calcul du résidu studentisé
res.student <- rstudent(reg)
#risque alpha = 0.1
alpha <- 0.1
#calcul du seuil à partir de la loi de Student à (n-p-2) ddl
seuil.student <- qt(1-alpha/2,31-4-2)
#construction du graphique des résidus standardisés
plot(autos$conso,res.student)
abline(h=-seuil.student)
abline(h+=seuil.student)
abline(h=0)
#détection des points en dehors des tuyaux
ab.student <- autos[res.student < -seuil.student | res.student > +seuil.student,]
#mettre en évidence les points atypiques dans le graphique
for (i in 1:nrow(ab.student)){
  vehicule <- row.names(ab.student)[i]
  text(autos[vehicule,"conso"],res.student[vehicule],vehicule)
}
```



Étude des points atypiques

Les points « leviers » (on ne tient compte que des exogènes via la hat-matrix)

```
#*****  
#autres mesures d'atypisme des points -- le levier  
#*****  
#un autre outil pour récupérer les indicateurs d'influence  
atypiques <- influence.measures(reg)  
#quels sont les descripteurs disponibles  
attributes(atypiques)  
#on s'intéresse à la matrice infmat  
print(atypiques$infmat)  
#on récupère la colonne "hat" qui correspond au levier  
res.hat <- atypiques$infmat[,"hat"]  
#le seuil est défini par  $2x(p+1)/n$   
seuil.hat <- 2*(4+1)/31  
#les points atypiques au sens du levier  
ab.hat <- autos[res.hat > seuil.hat,]  
print(ab.hat)
```



```
R Console  
> print(ab.hat)  
      prix cylindree puissance poids conso  
Ferrari 456 GT      285000      5474      325 1690 21.3  
Mercedes S 600     183900      5987      300 2250 18.7  
Maserati Ghibli GT  92500      2789      209 1485 14.5  
> |
```

Par rapport aux deux premiers indicateurs (résidu standardisé et résidu studentisé) :

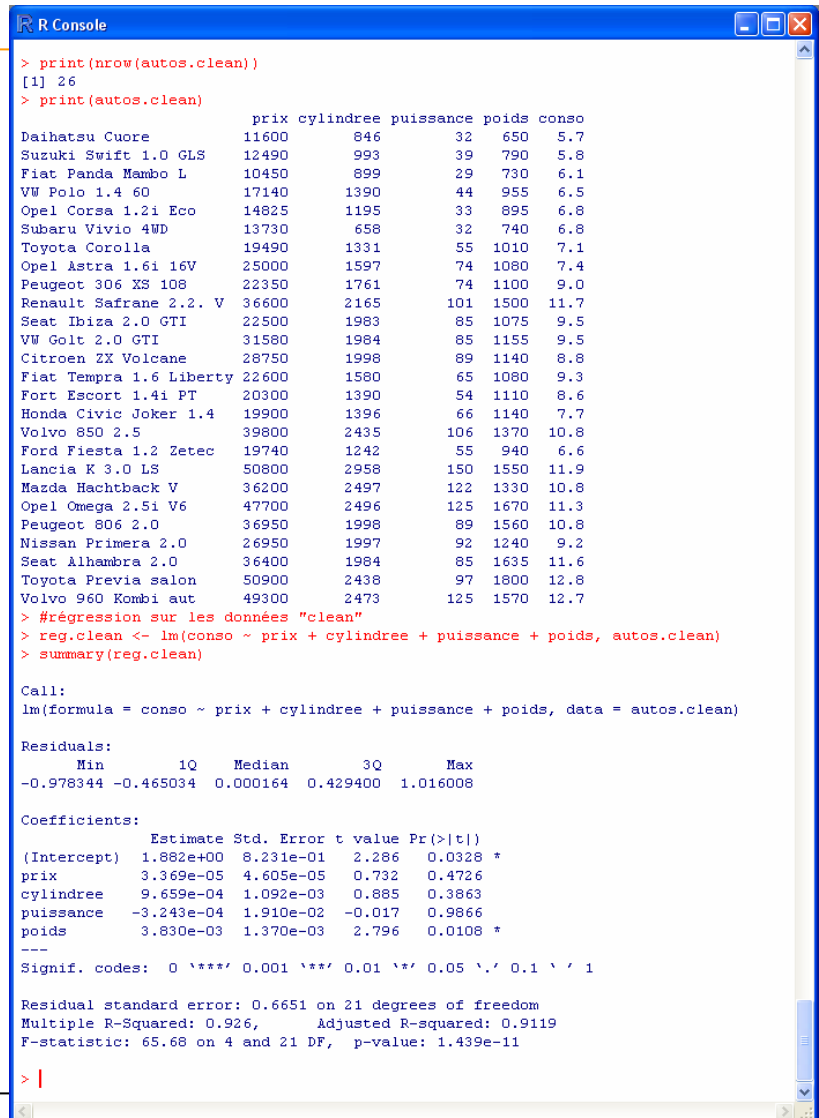
- (a) Ferrari et Mercedes restent atypiques, ils sont à la fois atypiques dans la description comme dans la prédiction de la consommation
- (b) Maserati apparaît comme atypique (sur les exogènes)
- (c) Mitsubishi et Hyundai n'ont pas une description très (significativement) différente des autres

Traitement des points atypiques

```
#vecteur booléen indicateur de suspicion pour résidu standardisé
b.standard <- (res.standard < -seuil.standard | res.standard > +seuil.standard)
#vecteur booléen indicateur de suspicion pour résidu studentisé
b.student <- (res.student < -seuil.student | res.student > +seuil.student)
#vecteur booléen indicateur de suspicion pour le levier
b.hat <- (res.hat > seuil.hat)
#booléen indicateur de détection au moins une fois
b.suspicious <- b.standard | b.student | b.hat
#booléen indicateur de non détection
b.not.suspicious <- !b.suspicious
#data.frame autos sans les données suspectes → 26 obs. a priori
autos.clean <- autos[b.not.suspicious,]
print(nrow(autos.clean))
print(autos.clean)
#régression sur les données "clean"
reg.clean <- lm(conso ~ prix + cylindree + puissance + poids, autos.clean)
summary(reg.clean)
```

Adoptons une règle drastique (et terriblement fruste) : un point au moins une fois atypique au sens des critères ci-dessus est éliminé de la base.

Alors que précédemment, prix, puissance et poids étaient significatifs à 5%, dans cette seconde régression, sans les 5 points « atypiques », seul **poids** semble peser réellement sur la consommation.



```
R Console
> print(nrow(autos.clean))
[1] 26
> print(autos.clean)
      prix cylindree puissance poids conso
Daihatsu Cuore      11600      846      32    650  5.7
Suzuki Swift 1.0 GLS 12490      993      39    790  5.8
Fiat Panda Mambo L  10450      899      29    730  6.1
VW Polo 1.4 60      17140     1390      44    955  6.5
Opel Corsa 1.2i Eco 14825     1195      33    895  6.8
Subaru Vivio 4WD    13730      658      32    740  6.8
Toyota Corolla      19490     1331      55   1010  7.1
Opel Astra 1.6i 16V 25000     1597      74   1080  7.4
Peugeot 306 XS 108  22350     1761      74   1100  9.0
Renault Safrane 2.2. V 36600     2165     101   1500 11.7
Seat Ibiza 2.0 GTI  22500     1983      85   1075  9.5
VW Golt 2.0 GTI     31580     1984      85   1155  9.5
Citroen ZX Volcane  28750     1998      89   1140  8.8
Fiat Tempra 1.6 Liberty 22600     1580      65   1080  9.3
Ford Escort 1.4i PT 20300     1390      54   1110  8.6
Honda Civic Joker 1.4 19900     1396      66   1140  7.7
Volvo 850 2.5       39800     2435     106   1370 10.8
Ford Fiesta 1.2 Zetec 19740     1242      55    940  6.6
Lancia K 3.0 LS     50800     2958     150   1550 11.9
Mazda Hachtback V  36200     2497     122   1330 10.8
Opel Omega 2.5i V6  47700     2496     125   1670 11.3
Peugeot 806 2.0     36950     1998      89   1560 10.8
Nissan Primera 2.0  26950     1997      92   1240  9.2
Seat Alhambra 2.0   36400     1984      85   1635 11.6
Toyota Previa salon 50900     2438      97   1800 12.8
Volvo 960 Kombi aut 49300     2473     125   1570 12.7
> #régression sur les données "clean"
> reg.clean <- lm(conso ~ prix + cylindree + puissance + poids, autos.clean)
> summary(reg.clean)

Call:
lm(formula = conso ~ prix + cylindree + puissance + poids, data = autos.clean)

Residuals:
    Min       1Q   Median       3Q      Max
-0.978344 -0.465034  0.000164  0.429400  1.016008

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.882e+00  8.231e-01  2.286  0.0328 *
prix         3.369e-05  4.605e-05  0.732  0.4726
cylindree    9.659e-04  1.092e-03  0.885  0.3863
puissance   -3.243e-04  1.910e-02 -0.017  0.9866
poids        3.830e-03  1.370e-03  2.796  0.0108 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6651 on 21 degrees of freedom
Multiple R-Squared: 0.926, Adjusted R-squared: 0.9119
F-statistic: 65.68 on 4 and 21 DF, p-value: 1.439e-11

> |
```

Détection de la colinéarité

Règle de Klein

```
#détection de la colinéarité - Règle de Klein
#calculer la matrice des corrélations croisées
mcxx <- cor(autos.clean[,c(1,2,3,4)])
#monter au carré
mcxx <- mcxx^2
#affichage
mcxx
```

Règle de Klein : il y a suspicion de colinéarité si le carré de la corrélation entre deux au moins des exogènes est proche du coefficient de détermination de la régression, égal à 0.926

```
R Console
> #affichage
> mcxx
      prix cylindree puissance  poids
prix    1.0000000  0.8727168  0.8550363  0.8952794
cylindree 0.8727168  1.0000000  0.9481205  0.7774552
puissance 0.8550363  0.9481205  1.0000000  0.7213830
poids     0.8952794  0.7774552  0.7213830  1.0000000
> |
```

La liaison entre puissance et cylindree pose problème...

```
R Console
> summary(reg.clean)

Call:
lm(formula = conso ~ prix + cylindree + puissance + poids, data = autos.clean)

Residuals:
    Min       1Q   Median       3Q      Max
-0.978344 -0.465034  0.000164  0.429400  1.016008

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.882e+00  8.231e-01   2.286  0.0328 *
prix         3.369e-05  4.605e-05   0.732  0.4726
cylindree    9.659e-04  1.092e-03   0.885  0.3863
puissance   -3.243e-04  1.910e-02  -0.017  0.9866
poids        3.830e-03  1.370e-03   2.796  0.0108 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6651 on 21 degrees of freedom
Multiple R-Squared: 0.926,    Adjusted R-squared: 0.9119
F-statistic: 65.68 on 4 and 21 DF,  p-value: 1.439e-11

> |
```

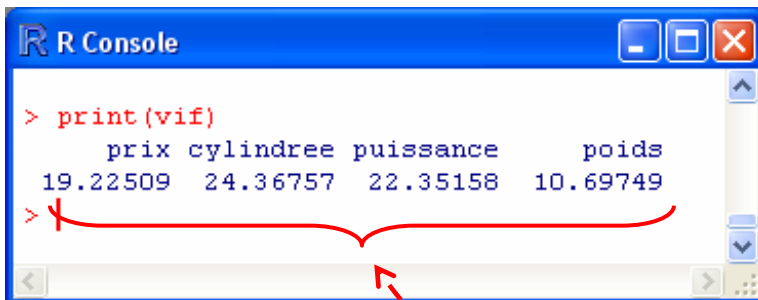
Détection de la colinéarité

Facteur d'inflation de la variance (VIF)

```
#préparer le vecteur de résultats des R2
r2 <- double(4) #4 parce qu'il y a 4 exogènes candidates

#pour chaque exogène
for (j1 in 1:4){
  #l'idée est de construire une chaîne de caractère représentant la formule
  str_formule <- paste(names(autos.clean[j1]), "~")
  for (j2 in 1:4){
    if (j2 != j1){
      str_formule <- paste(str_formule, names(autos.clean[j2]), "+")
    }
  }
  #enlever le dernier " +" à la fin de la chaîne
  str_formule <- substr(str_formule, 1, nchar(str_formule)-2)
  #transformer la chaîne en objet de type formule
  formule <- as.formula(str_formule)
  #lancer la régression
  regtest <- lm(formule, data=autos.clean)
  #récupérer le résumé
  resume.regtest <- summary(regtest)
  #et le R2 dans le résumé
  r2[j1] <- resume.regtest$r.squared
}

#calculer le VIF à partir du R2
vif <- 1/(1-r2)
#attribuer les noms des exogènes pour l'affichage
names(vif) <- names(autos.clean)[1:4]
#afficher
print(vif)
```



```
> print(vif)
      prix cylindree puissance      poids
19.22509 24.36757 22.35158 10.69749
```

Le VIF est une sorte de généralisation multivariée de la règle de Klein. Elle étudie la liaison, non plus entre les exogènes 2 à 2, mais entre chaque exogène et toutes les autres.

Le VIF de la variable X_j est déduit du coefficient de détermination de la régression de X_j avec tous les autres X . On considère qu'il y a un sérieux problème dès lors que $VIF > 10$.

Conclusion : il n'y a que des problèmes dans ce fichier !!!

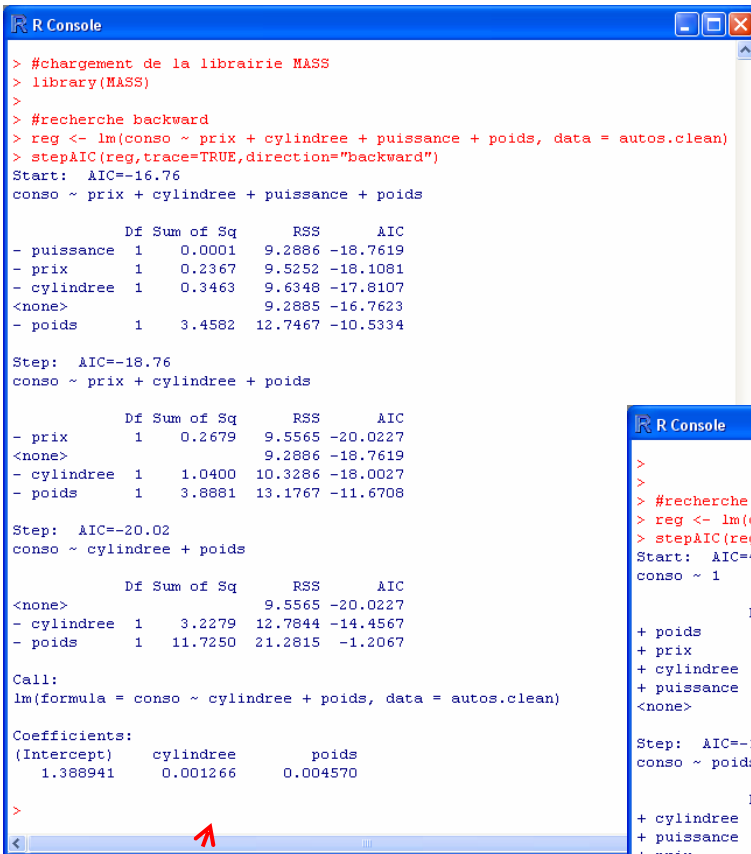
Selection de variables

Basée sur le critère AIC

```
#chargement de la librairie MASS
library(MASS)

#recherche backward
reg <- lm(conso ~ prix + cylindree + puissance + poids, data = autos.clean)
stepAIC(reg,trace=TRUE,direction="backward")

#recherche forward
reg <- lm(conso ~ 1,data=autos.clean)
stepAIC(reg,scope=list(upper=~prix+cylindree+puissance+poids,lower=~1),direction="
forward",trace=TRUE)
```



```
R Console
> #chargement de la librairie MASS
> library(MASS)
>
> #recherche backward
> reg <- lm(conso ~ prix + cylindree + puissance + poids, data = autos.clean)
> stepAIC(reg,trace=TRUE,direction="backward")
Start:  AIC=-16.76
conso ~ prix + cylindree + puissance + poids

      Df Sum of Sq  RSS    AIC
- puissance  1   0.0001  9.2886 -18.7619
- prix       1   0.2367  9.5252 -18.1081
- cylindree  1   0.3463  9.6348 -17.8107
<none>      0   9.2885 -16.7623
- poids     1   3.4582 12.7467 -10.5334

Step:  AIC=-18.76
conso ~ prix + cylindree + poids

      Df Sum of Sq  RSS    AIC
- prix       1   0.2679  9.5565 -20.0227
<none>      0   9.2886 -18.7619
- cylindree  1   1.0400 10.3286 -18.0027
- poids     1   3.8881 13.1767 -11.6708

Step:  AIC=-20.02
conso ~ cylindree + poids

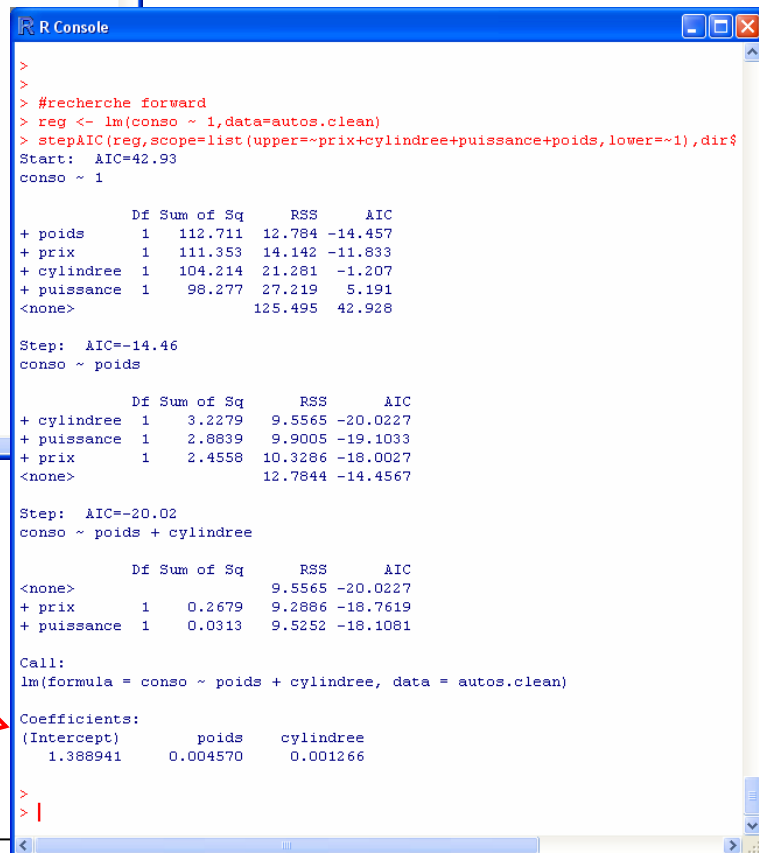
      Df Sum of Sq  RSS    AIC
<none>      0   9.5565 -20.0227
- cylindree  1   3.2279 12.7844 -14.4567
- poids     1  11.7250 21.2815  -1.2067

Call:
lm(formula = conso ~ cylindree + poids, data = autos.clean)

Coefficients:
(Intercept)  cylindree      poids
 1.388941    0.001266    0.004570
>
```

Le critère AIC (Akaike) met en balance la qualité de l'approximation (via le SCR ou RSS en anglais) et la complexité du modèle (via le nombre de variables).

Il permet de mettre en compétition des modèles avec un nombre de variables différent.



```
R Console
>
>
> #recherche forward
> reg <- lm(conso ~ 1,data=autos.clean)
> stepAIC(reg,scope=list(upper=~prix+cylindree+puissance+poids,lower=~1),dir$
Start:  AIC=42.93
conso ~ 1

      Df Sum of Sq  RSS    AIC
+ poids     1  112.711 12.784 -14.457
+ prix      1  111.353 14.142 -11.833
+ cylindree  1  104.214 21.281  -1.207
+ puissance  1   98.277 27.219   5.191
<none>      0  125.495 42.928

Step:  AIC=-14.46
conso ~ poids

      Df Sum of Sq  RSS    AIC
+ cylindree  1   3.2279  9.5565 -20.0227
+ puissance  1   2.8839  9.9005 -19.1033
+ prix       1   2.4558 10.3286 -18.0027
<none>      0  12.7844 -14.4567

Step:  AIC=-20.02
conso ~ poids + cylindree

      Df Sum of Sq  RSS    AIC
<none>      0   9.5565 -20.0227
+ prix     1   0.2679  9.2886 -18.7619
+ puissance  1   0.0313  9.5252 -18.1081

Call:
lm(formula = conso ~ poids + cylindree, data = autos.clean)

Coefficients:
(Intercept)      poids  cylindree
 1.388941    0.004570    0.001266
>
> |
```

Recherche « backward » et recherche « forward » aboutissent au même résultat : le poids et la cylindrée sont les variables pertinentes pour expliquer la consommation des véhicules

***Et on peut faire bien
d'autres choses encore...***