Sélection de modèle

Sélection de modèle sur le jeu de données entier

On charge le package *leaps* et le jeu de données de rétinol plasmatique

```
require(leaps)
```

```
## Loading required package: leaps
```

```
retinol <- read.csv2("~/codes/presentationTPretinol.csv", header = T)
names(retinol)
```

```
##  [1] "age"         "sexe"        "tabac"       "bmi"         "vitamine"
##  [6] "calories"    "graisses"    "fibres"      "alcool"      "cholesterol"
## [11] "betadiet"    "retdiet"     "betaplasma"  "retplasma"
```

```
regfit.full = regsubsets(retplasma~. , retinol)
summary(regfit.full)
```

```
## Subset selection object
## Call: regsubsets.formula(retplasma ~ ., retinol)
## 13 Variables  (and intercept)
##             Forced in Forced out
## age             FALSE      FALSE
## sexe            FALSE      FALSE
## tabac           FALSE      FALSE
## bmi             FALSE      FALSE
## vitamine        FALSE      FALSE
## calories        FALSE      FALSE
## graisses        FALSE      FALSE
## fibres          FALSE      FALSE
## alcool          FALSE      FALSE
## cholesterol     FALSE      FALSE
## betadiet        FALSE      FALSE
## retdiet         FALSE      FALSE
## betaplasma      FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          age sexe tabac bmi vitamine calories graisses fibres alcool
## 1  ( 1 ) "*" " "  " "   " " " "      " "      " "      " "    " "
## 2  ( 1 ) "*" "*"  " "   " " " "      " "      " "      " "    " "
## 3  ( 1 ) "*" "*"  " "   " " " "      " "      "*"      " "    " "
## 4  ( 1 ) "*" "*"  " "   " " " "      " "      "*"      " "    " "
## 5  ( 1 ) "*" "*"  " "   " " " "      " "      " "      "*"    " "
## 6  ( 1 ) "*" "*"  " "   " " "*"      " "      " "      "*"    " "
## 7  ( 1 ) "*" "*"  "*"   " " "*"      " "      " "      "*"    " "
## 8  ( 1 ) "*" "*"  "*"   "*" "*"      " "      " "      "*"    " "
##          cholesterol betadiet retdiet betaplasma
## 1  ( 1 ) " "         " "      " "     " "
## 2  ( 1 ) " "         " "      " "     " "
## 3  ( 1 ) " "         " "      " "     " "
## 4  ( 1 ) " "         " "      " "     "*"
## 5  ( 1 ) "*"         " "      " "     "*"
```

```
## 6  ( 1 ) "*"         " "       " "      "*"
## 7  ( 1 ) "*"         " "       " "      "*"
## 8  ( 1 ) "*"         " "       " "      "*"
```

On peut limiter le nombre de variables dans un modèle à tester via le paramètre *nvmax*

```
regfit.full = regsubsets(retplasma~., data = retinol, nvmax=13)
summary(regfit.full)
```

```
## Subset selection object
## Call: regsubsets.formula(retplasma ~ ., data = retinol, nvmax = 13)
## 13 Variables  (and intercept)
##             Forced in Forced out
## age             FALSE      FALSE
## sexe            FALSE      FALSE
## tabac           FALSE      FALSE
## bmi             FALSE      FALSE
## vitamine        FALSE      FALSE
## calories        FALSE      FALSE
## graisses        FALSE      FALSE
## fibres          FALSE      FALSE
## alcool          FALSE      FALSE
## cholesterol     FALSE      FALSE
## betadiet        FALSE      FALSE
## retdiet         FALSE      FALSE
## betaplasma      FALSE      FALSE
## 1 subsets of each size up to 13
## Selection Algorithm: exhaustive
##           age sexe tabac bmi vitamine calories graisses fibres alcool
## 1  ( 1 )  "*" " "  " "   " " " "      " "       " "       " "    " "
## 2  ( 1 )  "*" "*"  " "   " " " "      " "       " "       " "    " "
## 3  ( 1 )  "*" "*"  " "   " " " "      " "       "*"       " "    " "
## 4  ( 1 )  "*" "*"  " "   " " " "      " "       "*"       " "    " "
## 5  ( 1 )  "*" "*"  " "   " " " "      " "       " "       "*"    " "
## 6  ( 1 )  "*" "*"  " "   " " "*"      " "       " "       "*"    " "
## 7  ( 1 )  "*" "*"  "*"   " " "*"      " "       " "       "*"    " "
## 8  ( 1 )  "*" "*"  "*"   "*" "*"      " "       " "       "*"    " "
## 9  ( 1 )  "*" "*"  "*"   " " " "      "*"       "*"       "*"    "*"
## 10  ( 1 ) "*" "*"  "*"   "*" " "      "*"       "*"       "*"    "*"
## 11  ( 1 ) "*" "*"  "*"   "*" "*"      "*"       "*"       "*"    "*"
## 12  ( 1 ) "*" "*"  "*"   "*" "*"      "*"       "*"       "*"    "*"
## 13  ( 1 ) "*" "*"  "*"   "*" "*"      "*"       "*"       "*"    "*"
##           cholesterol betadiet retdiet betaplasma
## 1  ( 1 )  " "         " "      " "     " "
## 2  ( 1 )  " "         " "      " "     " "
## 3  ( 1 )  " "         " "      " "     " "
## 4  ( 1 )  " "         " "      " "     "*"
## 5  ( 1 )  "*"         " "      " "     "*"
## 6  ( 1 )  "*"         " "      " "     "*"
## 7  ( 1 )  "*"         " "      " "     "*"
## 8  ( 1 )  "*"         " "      " "     "*"
## 9  ( 1 )  "*"         " "      " "     "*"
## 10  ( 1 ) "*"         " "      " "     "*"
## 11  ( 1 ) "*"         " "      " "     "*"
## 12  ( 1 ) "*"         " "      "*"     "*"
```

```
## 13  ( 1 ) "*"         "*"      "*"     "*"
```

```
reg.summary = summary(regfit.full)
names(reg.summary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```
par(mfrow=c(2 ,2))
plot(reg.summary$rss , xlab =" Number of Variables " , ylab =" RSS " ,type ="l")
plot(reg.summary$adjr2 , xlab =" Number of Variables " , ylab =" Adjusted RSq " , type ="l")
which.max(reg.summary$adjr2)
```

```
## [1] 4
```

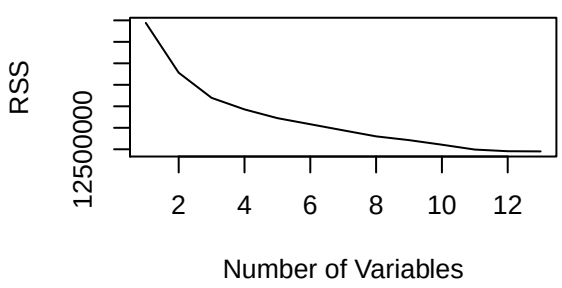
```
points(which.max(reg.summary$adjr2), reg.summary$adjr2[which.max(reg.summary$adjr2)], col =" red " , ce

plot(reg.summary$cp , xlab =" Number of Variables " , ylab =" Cp " ,type = "l")
which.max(reg.summary$cp)
```

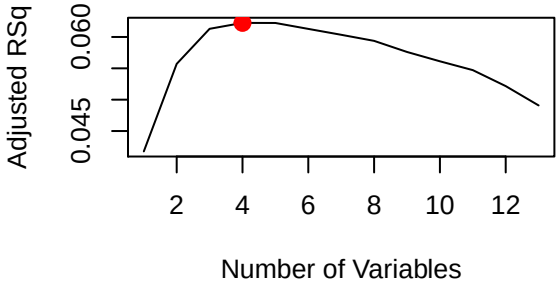
```
## [1] 13
```

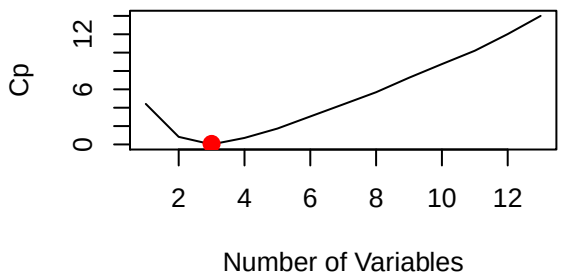
```
points(which.min(reg.summary$cp), reg.summary$adjr2[which.min(reg.summary$cp)], col =" red " , cex =2 ,

plot(reg.summary$bic , xlab =" Number of Variables " , ylab =" BIC " , type ="l")
which.min(reg.summary$bic)
```

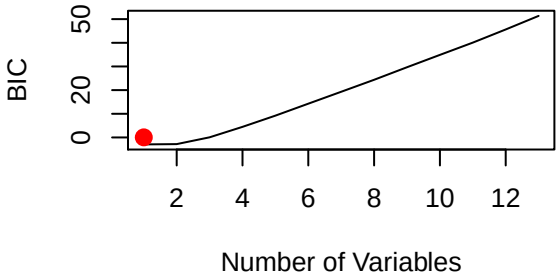
```
## [1] 1
```

```
points(which.min(reg.summary$bic), reg.summary$adjr2[which.min(reg.summary$bic)], col =" red " , cex =2
```





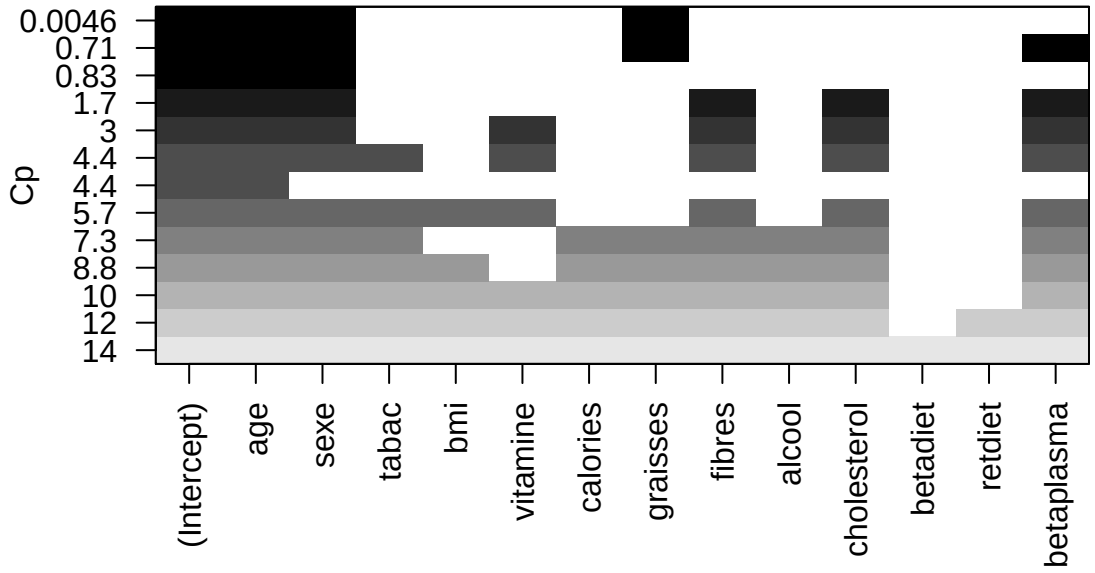




Le package *leap* possède sa propre fonction *plot*

```
#plot(regfit.full , scale ="r2")
#plot(regfit.full , scale ="bic")
#plot(regfit.full , scale ="adjr2")
```

```
plot(regfit.full , scale ="Cp")
```



On peut accéder aux coefficient de régression du modèle à 1 variable

```
coef(regfit.full , 1)
```

```
## (Intercept)         age
##  450.661090    3.033727
```

On peut faire aussi de la recherche pas à pas en *forward* et *backward*

```
regfit.fwd = regsubsets(retplasma~. , data = retinol , nvmax =13 , method ="forward")
summary(regfit.fwd)
```

```
## Subset selection object
## Call: regsubsets.formula(retplasma ~ ., data = retinol, nvmax = 13,
##     method = "forward")
## 13 Variables  (and intercept)
##             Forced in Forced out
## age             FALSE      FALSE
## sexe            FALSE      FALSE
## tabac           FALSE      FALSE
## bmi             FALSE      FALSE
## vitamine        FALSE      FALSE
## calories        FALSE      FALSE
## graisses        FALSE      FALSE
## fibres          FALSE      FALSE
## alcool          FALSE      FALSE
## cholesterol     FALSE      FALSE
## betadiet        FALSE      FALSE
## retdiet         FALSE      FALSE
## betaplasma      FALSE      FALSE
## 1 subsets of each size up to 13
## Selection Algorithm: forward
##          age sexe tabac bmi vitamine calories graisses fibres alcool
## 1  ( 1 )  "*" " "  " "   " " " "      " "      " "      " "    " "
## 2  ( 1 )  "*" "*"  " "   " " " "      " "      " "      " "    " "
```

```
## 3  ( 1 )  "*" "*"  " "   " " " "      " "      "*"      " "    " "
## 4  ( 1 )  "*" "*"  " "   " " " "      " "      "*"      " "    " "
## 5  ( 1 )  "*" "*"  " "   " " " "      " "      "*"      "*"    " "
## 6  ( 1 )  "*" "*"  " "   " " "*"      " "      "*"      "*"    " "
## 7  ( 1 )  "*" "*"  "*"   " " "*"      " "      "*"      "*"    " "
## 8  ( 1 )  "*" "*"  "*"   "*" "*"      " "      "*"      "*"    " "
## 9  ( 1 )  "*" "*"  "*"   "*" "*"      " "      "*"      "*"    " "
## 10  ( 1 ) "*" "*"  "*"   "*" "*"      "*"      "*"      "*"    " "
## 11  ( 1 ) "*" "*"  "*"   "*" "*"      "*"      "*"      "*"    "*"
## 12  ( 1 ) "*" "*"  "*"   "*" "*"      "*"      "*"      "*"    "*"
## 13  ( 1 ) "*" "*"  "*"   "*" "*"      "*"      "*"      "*"    "*"
##           cholesterol betadiet retdiet betaplasma
## 1  ( 1 )  " "         " "      " "     " "
## 2  ( 1 )  " "         " "      " "     " "
## 3  ( 1 )  " "         " "      " "     " "
## 4  ( 1 )  " "         " "      " "     "*"
## 5  ( 1 )  " "         " "      " "     "*"
## 6  ( 1 )  " "         " "      " "     "*"
## 7  ( 1 )  " "         " "      " "     "*"
## 8  ( 1 )  " "         " "      " "     "*"
## 9  ( 1 )  "*"         " "      " "     "*"
## 10  ( 1 ) "*"         " "      " "     "*"
## 11  ( 1 ) "*"         " "      " "     "*"
## 12  ( 1 ) "*"         " "      "*"     "*"
## 13  ( 1 ) "*"         "*"      "*"     "*"
```

```
regfit.bwd = regsubsets(retplasma~. , data = retinol , nvmax =13 , method ="backward")
summary(regfit.bwd)
```

```
## Subset selection object
## Call: regsubsets.formula(retplasma ~ ., data = retinol, nvmax = 13,
##     method = "backward")
## 13 Variables  (and intercept)
##             Forced in Forced out
## age             FALSE      FALSE
## sexe            FALSE      FALSE
## tabac           FALSE      FALSE
## bmi             FALSE      FALSE
## vitamine        FALSE      FALSE
## calories        FALSE      FALSE
## graisses        FALSE      FALSE
## fibres          FALSE      FALSE
## alcool          FALSE      FALSE
## cholesterol     FALSE      FALSE
## betadiet        FALSE      FALSE
## retdiet         FALSE      FALSE
## betaplasma      FALSE      FALSE
## 1 subsets of each size up to 13
## Selection Algorithm: backward
##           age sexe tabac bmi vitamine calories graisses fibres alcool
## 1  ( 1 )  "*" " "  " "   " " " "      " "      " "      " "    " "
## 2  ( 1 )  "*" "*"  " "   " " " "      " "      " "      " "    " "
## 3  ( 1 )  "*" "*"  " "   " " " "      " "      "*"      " "    " "
## 4  ( 1 )  "*" "*"  " "   " " " "      " "      "*"      " "    " "
## 5  ( 1 )  "*" "*"  " "   " " " "      " "      "*"      "*"    " "
```

```
## 6  ( 1 )  "*" "*"  " "   " " " "      "*"      "*"      "*"    " "
## 7  ( 1 )  "*" "*"  " "   " " " "      "*"      "*"      "*"    "*"
## 8  ( 1 )  "*" "*"  "*"   " " " "      "*"      "*"      "*"    "*"
## 9  ( 1 )  "*" "*"  "*"   " " " "      "*"      "*"      "*"    "*"
## 10  ( 1 ) "*" "*"  "*"   "*" " "      "*"      "*"      "*"    "*"
## 11  ( 1 ) "*" "*"  "*"   "*" "*"      "*"      "*"      "*"    "*"
## 12  ( 1 ) "*" "*"  "*"   "*" "*"      "*"      "*"      "*"    "*"
## 13  ( 1 ) "*" "*"  "*"   "*" "*"      "*"      "*"      "*"    "*"
##           cholesterol betadiet retdiet betaplasma
## 1  ( 1 )  " "         " "      " "     " "
## 2  ( 1 )  " "         " "      " "     " "
## 3  ( 1 )  " "         " "      " "     " "
## 4  ( 1 )  " "         " "      " "     "*"
## 5  ( 1 )  " "         " "      " "     "*"
## 6  ( 1 )  " "         " "      " "     "*"
## 7  ( 1 )  " "         " "      " "     "*"
## 8  ( 1 )  " "         " "      " "     "*"
## 9  ( 1 )  "*"         " "      " "     "*"
## 10  ( 1 ) "*"         " "      " "     "*"
## 11  ( 1 ) "*"         " "      " "     "*"
## 12  ( 1 ) "*"         " "      "*"     "*"
## 13  ( 1 ) "*"         "*"      "*"     "*"
```

Sélection de modèle avec des ensembles “apprentissage-validation” et validation croisée

On construit les deux ensembles *apprentissage validation* aléatoirement

```
set.seed(1)
train = sample (c(TRUE,FALSE) , nrow(retinol) , rep = TRUE)
test=(!train)
```

```
regfit.best = regsubsets(retplasma~. , data = retinol[train,],nvmax=13)
```

On peut récupérer la matrice dite de “design” du modèle complet sur les données de *validation*

```
test.mat = model.matrix(retplasma~. , data = retinol[test,])
```

On va maintenant calculer l’erreur de prédiction sur l’ensemble de validation pour chaque nombre de variables dans le modèle

```
val.errors = rep(NA ,13)
for( i in 1:13){
  coefi = coef(regfit.best,  id=i)
  pred = test.mat[ ,names(coefi)]%*%coefi
  val.errors[i]=mean((retinol$retplasma[test] - pred)^2)
}
val.errors
```

```
##  [1] 43771.14 45740.63 46665.44 47107.73 46834.66 47707.43 47613.93
##  [8] 47297.19 46175.60 46066.54 45964.62 45856.02 45786.53
```

```
which.min(val.errors)
```

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

```
coef(regfit.full, 1)
```

```
## (Intercept)         age
##  450.661090    3.033727
```

On a besoin d'une fonction qui prédit à partir d'un objet *regsubset*

```
predict.regsubsets = function(object , newdata , id ,...){
form = as.formula(object$call[[2]])
mat = model.matrix( form , newdata)
coefi = coef(object , id=id)
xvars = names(coefi)
mat[ ,xvars]%*% coefi
}
```

On va faire une validation croisée à 5 *folds*

```
k=5
set.seed(1)
folds = sample(1:k, nrow(retinol) , replace = TRUE)
cv.errors = matrix(NA, k, 13 ,dimnames = list(NULL , paste(1:13)))
```

```
for(j in 1:k){
  best.fit = regsubsets(retplasma~. , data=retinol[folds!=j,] , nvmax=13)
  for (i in 1:13){
   pred = predict(best.fit , retinol[folds==j,] , id=i)
   cv.errors[j,i]= mean((retinol$retplasma[folds==j]-pred) ^2)
   }
 }
```

La moyenne de l'erreur sur les 5 folds

```
mean.cv.errors = apply (cv.errors ,2 ,mean)
mean.cv.errors
```

```
##        1        2        3        4        5        6        7        8
## 44007.03 57082.60 55153.92 54947.82 55262.78 55872.08 56432.02 56329.78
##        9       10       11       12       13
## 55839.44 56012.73 55787.21 55686.20 55891.40
```