TD Manipulation des données et apprentissage statistique

Magali Berland
magali.berland@uvsq.fr
Option "Introduction à la Modélisation en Biologie" M2 IHPS

4 décembre 2017

Table des matières

1 Les données

Nous allons utiliser les données `Prostate` du package `lasso2` de R. L'antigène prostatique spécifique (PSA) est une protéine fabriquée exclusivement par la prostate. Le PSA est normalement présent dans le sang de tous les hommes à un taux infime. Le dosage de son taux sanguin est utilisé pour le diagnostic et le suivi du cancer de la prostate. Un taux élevé de PSA peut suggérer un cancer.

Ces données viennent d'une étude qui a examiné la corrélation entre le niveau de PSA (`lpsa`) et huit mesures cliniques chez 97 hommes qui étaient sur le point de subir une prostatectomie.

```
library(ggfortify)
library(lasso2)
data(Prostate)
set.seed(103)
?Prostate
summary(Prostate)
```

2 Entrainement à la manipulation des données

1. Transformez les données de manière adéquate afin de pouvoir produire un boxplot de toutes les variables quantitatives
2. Trouvez tous les patients qui :
   — ont plus de 70 ans
   — ont un score gleason de 6 ou 7
   — ont un niveau d'antigène (`lpsa`) élevé (> 3) mais qui ne sont pas très âgés (< 45 ans)
   — ont entre de 50 et 55 ans (utilisez la fonction `between()`)
3. Triez les patients du plus grave au moins grave (`lpsa`)

4. Triez maintenant en fonction des variables `lcavol` et `lweight`. Trouvez-vous le même ordre ?
5. Groupez les patients par score gleason
6. Produisez un résumé qui calcule l'effectif du groupe, la moyenne et l'écart-type des autres variables

3 Modèle linéaire

3.1 Vérification des données et description élémentaire

1. Procédez à une exploration des données unidimensionnelle, bidimensionnelle et multivariée.

3.2 Construction du découpage des données

Considérons le partage des données suivant :

```
ind.test = sample(nrow(Prostate), size = round(nrow(Prostate)*0.25))
Prostate.app = Prostate[-ind.test,]
Prostate.test = Prostate[c(ind.test),]
```

3.3 Estimation du modèle linéaire complet

```
modlin = lm(lpsa ~ ., data = Prostate.app)
summary(modlin)
```

```
Call:
lm(formula = lpsa ~ ., data = Prostate.app)

Residuals:
     Min       1Q   Median       3Q      Max
-1.61568 -0.46823 -0.05194  0.36447  1.65938

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.519717   1.468553   0.354  0.72458
lcavol       0.589068   0.110671   5.323  1.4e-06 ***
lweight      0.387802   0.193772   2.001  0.04960 *
age         -0.015507   0.013066  -1.187  0.23968
lbph         0.108928   0.071991   1.513  0.13518
svi          0.788945   0.285840   2.760  0.00753 **
lcp         -0.091108   0.102545  -0.888  0.37761
gleason      0.071919   0.188413   0.382  0.70394
pgg45        0.003346   0.005148   0.650  0.51808
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7407 on 64 degrees of freedom
Multiple R-squared:  0.6434,        Adjusted R-squared:  0.5988
F-statistic: 14.43 on 8 and 64 DF,  p-value: 8.585e-12
```

Residuals : Caractéristiques numériques de la distribution des résidus. Pour que le modèle soit valide, les résidus doivent être symétriques par rapport à la moyenne, leur médiane doit être proche de 0 et les valeurs 1Q et 3Q doivent être de magnitude similaire.

Coefficients : L'estimation des coefficients du modèle (`Estimate`), leur erreur-type (`Std. Error`), un test pour déterminer s'ils sont significativement différent de 0. L'hypothèse nulle de ce test est "H0 : le coefficient du modèle est nul", et cette hypothèse peut être rejetée lorsque la p-value (`Pr(>|t|)`) est inférieure à un seuil que l'on peut fixer à 0.05 (5%).

Residual standard error : L'erreur standard des résidus donne une estimation de la qualité du modèle. Dans notre cas, le modèle que nous avons bâti prédit le niveau de PSA avec une erreur d'environ 0.71. Si les résidus suivent approximativement une loi normale, 2/3 des erreurs seront dans la plage ± 0.71 et 95 % dans ± 1.41.

Multiple R-squared, Adjusted R-squared : R^2 (coefficient de détermination), et R^2 ajusté par rapport au nombre de variables dans le modèle.

F-statistic : F est le ratio de la variance expliquée par la modèle et de la variance des résidus (variance inexpliquée). L'hypothèse nulle de ce test est "H0 : tous les paramètres (coefficients) de ce modèle sont nuls", l'hypothèse alternative est "H1 : il existe au moins un coeffcient qui n'est pas nul". Cela permet de déterminer si le modèle de régression explique une part significative de la variance totale.

3.4 Vérification des conditions d'application du modèle

```
autoplot(modlin, label.size = 3)
```

2. Pensez-vous que les conditions du modèle sont respectées ?

3. Calculez l'erreur d'apprentissage (moyenne de la différence au carré entre valeurs prédites et valeurs mesurées). Vous devez obtenir :

```
[1] 0.4810166
```

4 Sélection de variables

4.1 Sélection par AIC et backward

```
library(MASS)
modselect_b = stepAIC(modlin, ~., trace=TRUE, direction=c("backward"))
summary(modselect_b)
# noter que des paramètres restent non significatifs
```

4.2 Sélection par AIC et forward

```
mod0 = lm(lpsa ~ 1, data = Prostate)
modselect_f = stepAIC(mod0, lpsa ~ lcavol + lweight + age + lbph + svi +
                        lcp + gleason + pgg45, data = Prostate,
                        trace = TRUE, direction=c("forward"))
summary(modselect_f)
```

4.3 Sélection par AIC et stepwise

```
modselect = stepAIC(modlin, ~., trace = TRUE, direction = c("both"))
#both est l'option par défaut
summary(modselect)
```

4.4 Sélection par BIC et stepwise

```
modselect_BIC = stepAIC(modlin, ~., trace = TRUE, direction = c("both"),
                        k = log(length(Prostate$lpsa)))
summary(modselect_BIC)
```

4. Quel est le modèle le plus parcimonieux ?

4.5 Calcul de l'erreur du modèle

5. Calculez les erreurs de prédictions pour chacun des modèles précédent (sur les données mises de côté au début, `Prostate.test`). Quel est le meilleur modèle ?