Modèle linéaire, M1 EURIA

M1 Euria

Pierre Ailliot

Année 2024-2025

Contents

1 Introduction aux méthodes de régression

L'objectif général de la régression est d'expliquer une variable Y, dite réponse, variable exogène ou **variable à expliquer**, en fonction de p variables, dites endogènes ou **variables explicatives**. Il s'agit d'un des problèmes les plus classiques en statistique/machine learning/data science/intelligence artificielle. L'utilisation la plus usuelle en actuariat est liée à la **tarification** des contrats d'assurance. On cherche alors à prévoir le nombre,le montant ou la présence de sinistres d'un assuré en fonction de ses caractéristiques. En pratique, on dispose d'observations de ces variables sur n individus, c'est à dire d'un tableau de données de la forme :

y_1	$x_{1,1}$	$x_{1,2}$	$\dots$	$x_{1,p}$
y_2	$x_{2,1}$	$x_{2,2}$	$\dots$	$x_{2,p}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
y_n	$x_{n,1}$	$x_{n,2}$	$\dots$	$x_{n,p}$

Table 1: Lignes : individus, Colonnes : variables

La première colonne est la variable à prédire à partir des variables explicatives données dans les p dernières colonnes.

Ces différentes variables peuvent être

- **quantitatives à valeurs continues** (ex : âge, distance parcouru, montant...)
- **quantitatives à valeurs discrètes**, par exemple à valeurs binaires (ex : présence/absence d'une maladie, d'une fraude ou d'une vente) ou entières (ex : nombre de sinistres)
- **qualitatives** (ex : CSP, région, genre).

La nature des variables conditionne fortement le modèle de régression utilisé :

- dans les chapitres 2->3.3, toutes les variables seront supposées être quantitatives continues et on étudiera certains aspects du modèle de **régression linéaire gaussien**,
- dans les chapitres 3.4, on supposera toujours que la variable à expliquer est quantitative continue, mais on autorisera certaines variables explicatives à être qualitatives ou quantitatives discrètes et on introduira **l'analyse de la variance** et **l'analyse de la covariance**,
- enfin, les **modèles linéaires généralisés (GLM)**, abordés dans le chapitre 4, permettent de généraliser le modèle de régression linéaire lorsque la variable à expliquer est qualitative, à valeurs positives ou discrète.

D'autres modèles de régression non-linéaires (réseaux de neurones, forêts aléatoires, SVM,...) seront abordés dans le cours d'apprentissage statistique. Le cadre du modèle linéaire gaussien permet de simplifier l'inférence statistique : on peut calculer explicitement la loi des estimateurs, et en déduire des intervalles de confiance ou des tests d'hypothèses. La plupart de ces résultats se généralisent aux GLMs, mais plus on complexifie le modèle et plus les méthodes d'inférence statistique disponibles sont limitées. Une bonne compréhension des modèles linéaires puis linéaires généralisés est utile avant d'aborder les modèles plus complexes.

Les modèles GLM sont les modèles les plus utilisés en tarification : ils ont généralement des performances similaires aux modèles plus complexes, avec des temps de calculs plus faibles, de meilleures propriétés statistiques et surtout une plus grande interprétabilité.

Prérequis :

- Langage R pour les applications sur les données en TD/TP (cf polycopié du cours de L3 sur la page moodle du cours).
- Mathématiques : vecteurs gaussiens, projections orthogonales, lois du χ^2, de Student et de Fisher, théorème de Cochran (cf polycopié sur la page moodle du cours).
- Cours sur le modèle linéaire de L3 (cf polycopié sur la page moodle du cours).
- Cours de statistique de L3.

2 Rappels sur le modèle linéaire

Les points rappelés rapidement dans ce paragraphe ont été détaillés en L3 et sont considérés comme des prérequis pour ce cours (cf polycopié du cours de L3, disponible sur la page moodle du cours).

Définition 1 *Le **modèle de régression linéaire multiple** s'écrit*

$$Y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \dots \beta_p x_{i,p} + W_i$$

avec $(W_1, ..., W_n)$ *des **variables aléatoires** i.i.d. vérifiant* $E[W_i] = 0$ *et* $var(W_i) = \sigma^2 < \infty$ *et* $(\beta_0, ..., \beta_p, \sigma) \in \mathbb{R}^{p+1} \times \mathbb{R}^+$ *des **paramètres inconnus**.*

W est généralement appelé le vecteur de **résidus**. Les variables explicatives stockées dans la matrice X sont supposées déterministes alors que la variable à expliquer y_i est la réalisation d'une variable aléatoire Y_i. Le modèle se récrit sous la forme matricielle

$$Y = X\beta + W$$

avec $Y = (Y_1, ..., Y_n)'$, $\beta = (\beta_0, \beta_1, ..., \beta_p)'$, $W = (W_1, ..., W_n)'$ et

$$X = \begin{pmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,p} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \dots & x_{n,p} \end{pmatrix}$$

Quelques remarques sur le modèle de régression linéaire multiple :

- Lorsque $p = 0$, on a $Y_i = \beta_0 + W_i$ et les variables aléatoires $(Y_1, ..., Y_n)$ sont alors i.i.d. avec $E[Y_i] = \beta_0$ et $var(Y_i) = \sigma^2$. Le modèle de régression linéaire inclut donc le cas particulier des échantillons i.i.d. vu dans le cours de statistique en L3.
- Lorsque $p = 1$, le modèle est appelé 'modèle de régression linéaire simple' et lorsque $p > 1$ on parle de 'régression linéaire multiple'.
- Le coefficient β_0 est appelé 'Intercept'. Il est présent par défaut dans tous les modèles considérés dans ce cours. Il est associé à la colonne de 1 dans la matrice X.
- Toutes les variables explicatives sont supposés être quantitatives. La gestion des variables qualitatives sera discutée dans la suite du cours.

La **méthode des moindres carrés** est généralement utilisée pour estimer le vecteur β. **A retenir :**

- L'estimateur des moindres carrés de β est
$$\hat{B} = (X'X)^{-1}X'Y$$
avec X' la transposée de la matrice X.
- $X'X$ est une matrice symétrique qui est inversible si et seulement si la matrice X est de rang $p + 1$ (pas d'information redondante dans les covariables).
- $\hat{Y} = X\hat{B}$ est la projection orthogonale de Y sur $E = Im(X) = \{Xb | b \in \mathbb{R}^{p+1}\}$.
- $\hat{W} = Y - \hat{Y}$ est appelé **résidu empirique**.
- $\hat{B}$ est un estimateur sans biais de β et $S^2 = \frac{1}{n-(p+1)}\|\hat{W}\|^2$ est un estimateur sans biais de σ^2
- Il ne faut pas confondre l'**estimateur** $\hat{B}$ (qui est une variable aléatoire), l'**estimation** $\hat{b}$ qui est un vecteur numérique et la **vraie valeur des paramètres** β.

Afin de pouvoir construire des IC ou réaliser des tests d'hypothèse, on suppose généralement que les résidus suivent une loi normale. On parle alors de **modèle de régression linéaire multiple gaussien**.

Définition 2 *Le **modèle de régression linéaire multiple gaussien** s'écrit*

$$Y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \dots \beta_p x_{i,p} + W_i$$

avec $(W_1, ..., W_n) \sim_{iid} \mathcal{N}(0, \sigma^2)$.

Sous forme matricielle, sous les hypothèses du modèle linéaire gaussien, on a $W \sim \mathcal{N}(0, \sigma^2 I_n)$ et donc Y est un **vecteur gaussien**

$$Y = X\beta + W \sim \mathcal{N}(X\beta, \sigma^2 I_n).$$

L'étude des propriétés de l'estimateur des moindres carrés est donc liée aux propriétés des projections orthogonales de vecteur gaussien, et donc au **théorème de Cochran**. La proposition ci-dessous donne la loi des estimateurs des moindres carrés dans le modèle linéaire gaussien. Ces résultats permettent en particulier de construire des **intervalles de confiance** et de faire des **tests statistiques** sur la valeur des paramètres.

Proposition 1 *Sous les hypothèses du modèle linéaire gaussien, on a les propriétés suivantes :*

- $\hat{B} \sim \mathcal{N}(\beta, \sigma^2 (X'X)^{-1})$
- $(n-p-1)\frac{S^2}{\sigma^2}$ *suit une loi* χ^2_{n-p-1} *indépendante de* $\hat{B}$ *et* $\hat{Y}$.

Notons $H = (X'X)^{-1}$, $diag(H) = (h_{0,0}, ..., h_{p,p})$ *les éléments de la diagonale de la matrice* H, $\sigma^2(\hat{B}_i) = \sigma^2 h_{i,i}$ *et* $S^2(\hat{B}_i) = S^2 h_{i,i}$, *on a alors*

- $\hat{B}_i \sim \mathcal{N}(\beta_i, \sigma^2(\hat{B}_i))$,
- $\frac{\hat{B}_i - \beta_i}{S(\hat{B}_i)} \sim \mathcal{T}_{n-p-1}$.

Une fois le modèle ajusté, on peut l'utiliser pour faire des **prévisions**. Un des avantages du modèle linéaire gaussien est la possibilité de quantifier les incertitudes, soit sous la forme d'**intervalles de confiance** (quand on s'intéresse uniquement à la moyenne de la prédiction), soit sous la forme d'**intervalle de prédiction** (quand on s'intéresse à la valeur que va prendre la variable à expliquer).

Il ne faut pas oublier de **valider le modèle** avant de l'utiliser. La qualité globale du modèle peut se mesurer à l'aide du R^2 (parfois appelé coefficient de corrélation multiple) défini par

$$R^2 = \frac{\|\hat{Y} - \bar{Y}\|^2}{\|Y - \bar{Y}\|^2}$$

avec $\bar{Y} = \frac{1}{n}\sum_{i=1}^n Y_i$ la moyenne empirique de la variable à expliquer. A noter que $\bar{Y}$ s'interprète comme la meilleure prédiction au sens de moindres carrés de Y_i en l'absence de variables explicatives : on l'appelle parfois la prédiction 'naïve'. On vérifie facilement de $R^2 = 0$ si $\bar{Y} = \hat{Y}$, c'est à dire si le modèle ajusté n'apporte pas d'information supplémentaire par rapport au modèle 'naïf'. A l'opposé, si $R^2 = 1$, alors $Y = \hat{Y}$ et le modèle s'ajuste parfaitement aux données. De manière générale, $R^2 \in [0, 1]$ s'interprète comme la proportion de variation totale expliquée par le modèle.

Afin de valider plus précisément le modèle ajusté, on peut regarder les résidus. Sous les hypothèses du modèle linéaire, on doit avoir

$$(W_1, ..., W_n) \sim_{iid} \mathcal{N}(0, \sigma^2).$$

Cependant, les résidus W_i ne sont pas directement observés, la validation repose généralement sur les résidus empiriques $(\hat{W}_1, ..., \hat{W}_n)$. Les résidus empiriques ne sont pas des variables gaussiennes indépendantes sous les hypothèses du modèle linéaire, et une transformation ('standardisation') est généralement appliquée avant d'analyser les résidus.

Illustration avec R En pratique, on utilisera R pour ajuster les modèles de régression dans ce cours. Un exemple basé sur le jeu de données *iris* disponible dans R est donné ci-dessous.

```r
#Ajustement d'un modèle de régression linéaire simple
fit=lm(Sepal.Length~Petal.Length,data=iris)
#iris est le jeu de données sur lequel on travaille (disponible dans R)
#Sepal.Length est la variable à expliquer
#Sepal.Width est la variable explicative
confint(fit)  #intervalles de confiance pour beta
```

```
##                 2.5 %    97.5 %
## (Intercept)  4.1516972 4.4615096
## Petal.Length 0.3715907 0.4462539
```

```r
summary(fit)  #résumé ajustement
```

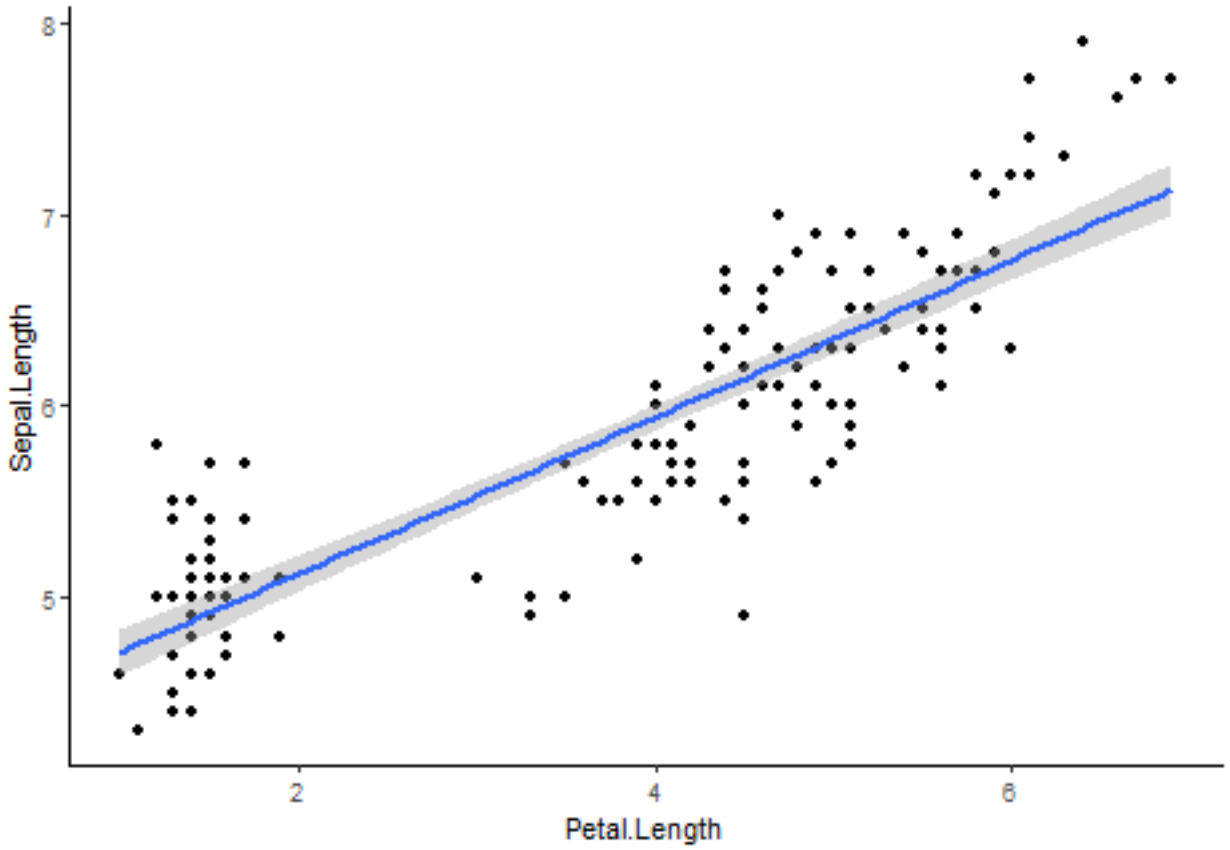
```
##
## Call:
## lm(formula = Sepal.Length ~ Petal.Length, data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.24675 -0.29657 -0.01515  0.27676  1.00269
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.30660    0.07839   54.94   <2e-16 ***
## Petal.Length  0.40892    0.01889   21.65   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4071 on 148 degrees of freedom
## Multiple R-squared:   0.76,  Adjusted R-squared:  0.7583
## F-statistic: 468.6 on 1 and 148 DF,  p-value: < 2.2e-16
```

```r
#Il faut savoir interpréter les sorties de la fonction summary.lm
#La partie 'Coefficients' donne en particulier les p-values des tests d'hypothèses H_0: beta_i=0
#La ligne 'Residual standard error' donne une estimation de sigma
#On peut en déduire des tests ou des intervalles de confiance pour sigma
#Le 'Multiple R-squared' mesure la qualité globale du modèle
#Les quantités 'Adjusted R-squared' et 'F-statistic' seront définis dans la suite du cours

#Représentation graphique
library(ggplot2)  #utilisation de ggplot2
```

```
## Warning: le package 'ggplot2' a été compilé avec la version R 4.2.2
```

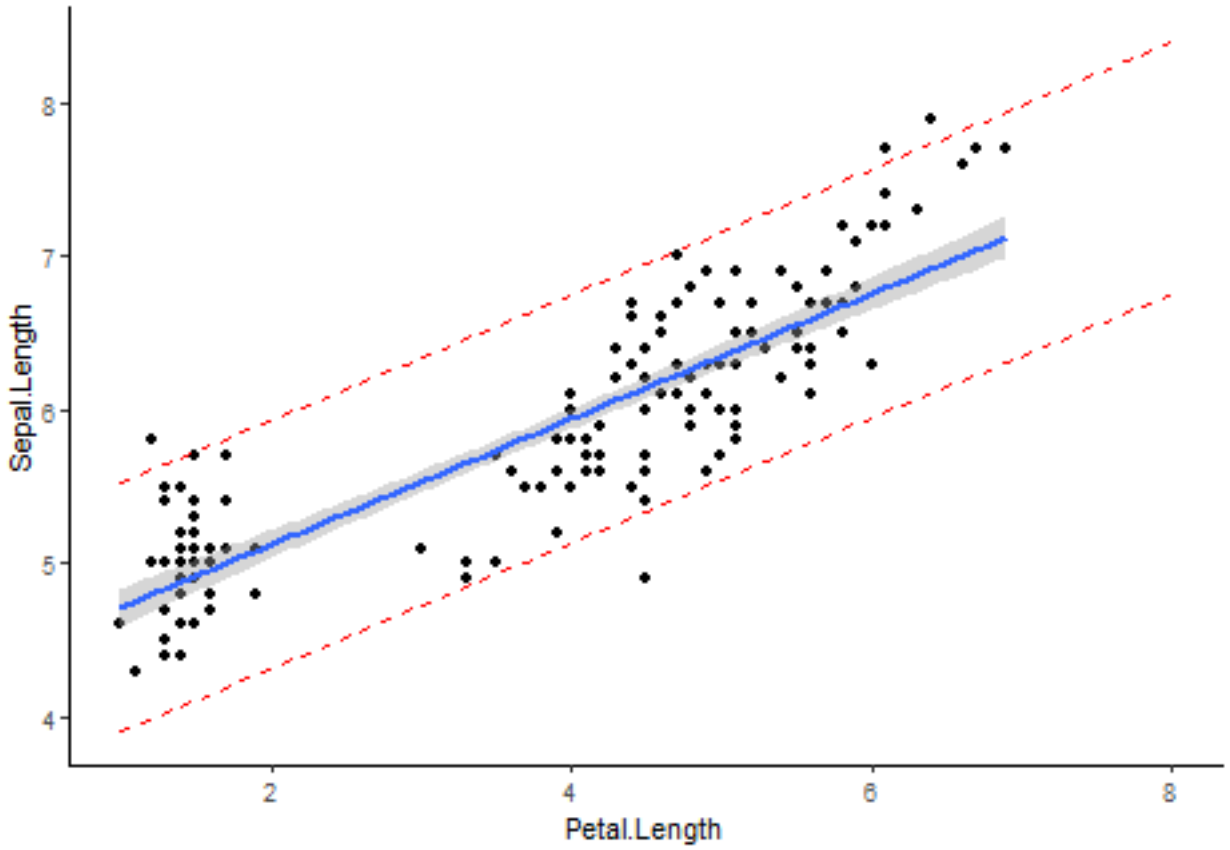
```r
ggplot(iris, aes(y=Sepal.Length, x=Petal.Length))+
  geom_point()+
  geom_smooth(method = "lm")+
  theme_classic()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

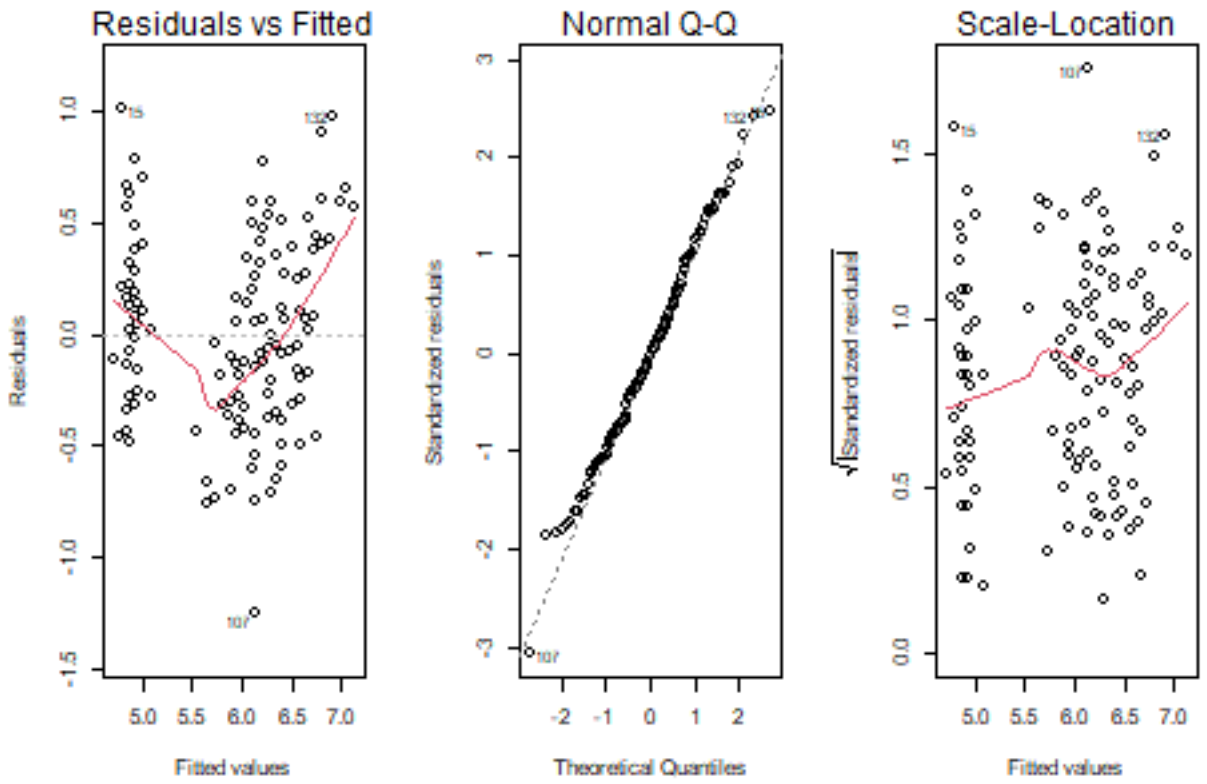



```
#la zone grisée est une intervalle de confiance à 95% pour la droite de régression

#Intervalle de prédicition
newx=data.frame(Petal.Length=seq(1,8,by=.1))
#création d'un data.frame qui contient les nouvelles valeurs de la variable explicative
#il faut que le nom des variables coincide avec le nom du jeu de données d'origine
pred=predict(fit,newx,interval = "prediction") #prédiction
pred=as.data.frame(cbind(newx,pred)) #stockage dans un data.frame
ggplot(iris, aes(y=Sepal.Length, x=Petal.Length))+
  geom_point()+
  geom_smooth(method = "lm")+
  geom_line(data=pred,aes(y=lwr, x=Petal.Length), color = "red", linetype = "dashed")+
 geom_line(data=pred,aes(y=upr, x=Petal.Length), color = "red", linetype = "dashed")+
  theme_classic()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
#le trait bleue est une prédiction ponctuelle (sans incertitude)
#la zone grisée représente un intervalle de confiance à 95% pour la moyenne
#les pointillés rouge donnent un intervalle de prédiction à 95%
#A retenir : on doit avoir environ 95% des observations dans l'IP à 95%!

#Validation du modèle
par(mfrow=c(1,3))
plot(fit,which=1:3)
```




```
#Il faut savoir interpréter les trois graphiques proposés par R
#Le premier graphique permet de vérifier si la relation est linéaire
#->points 'répartis dans une bande horizontale'?
#Le deuxième graphique permet de vérifier si le résidu est gaussien
#->points proches de la droite de Henry?
#Le dernier graphique permet de vérifier si le résidu est indépendant de x
#->points 'répartis dans une bande horizontale'?

#Ajustement d'un modèle de régression linéaire multiple
fit2=lm(Sepal.Length~.,data=iris[,1:4])
#Sepal.Length est la variable à expliquer
#La variable Species est enlevée (variable qualitative)
#Toutes les autres variables sont utilisées comme variables explicatives
summary(fit2)  #résumé ajustement
```

```
##
## Call:
## lm(formula = Sepal.Length ~ ., data = iris[, 1:4])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.82816 -0.21989  0.01875  0.19709  0.84570
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.85600    0.25078   7.401 9.85e-12 ***
## Sepal.Width   0.65084    0.06665   9.765  < 2e-16 ***
## Petal.Length  0.70913    0.05672  12.502  < 2e-16 ***
## Petal.Width  -0.55648    0.12755  -4.363 2.41e-05 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3145 on 146 degrees of freedom
## Multiple R-squared:  0.8586, Adjusted R-squared:  0.8557
## F-statistic: 295.5 on 3 and 146 DF,  p-value: < 2.2e-16
```

```
#toutes les variables explicatives ont un effet significatif
#le R^2 a augmenté par rapport au modèle de régression linéaire simple
```

3 Compléments sur le modèle linéaire

Dans ce chapitre, nous allons étudier plus en détails certains aspects du modèle linéaire.

3.1 Analyse de la variance de la régression

On a vu dans le cours de L3 comment tester l'hypothèse

$$H_0 : \beta_i = 0 \text{ contre } H_1 : \beta_i \neq 0$$

pour un coefficient quelconque $i \in \{0, ..., p\}$, en utilisant un test basé sur la loi de Student.

Plus généralement, le test d'analyse de la variance permet de tester une hypothèse de la forme

$$H_0 : \beta_1 = 0, \beta_2 = 0, ..., \beta_q = 0 \text{ contre } H_1 : \exists i \in \{1...q\},\ \beta_i \neq 0.$$

Quitte à renuméroter les variables, ceci permet de tester si un ensemble de $q \geq 1$ variables peut être enlevé des variables explicatives simultanément.

Dans la suite du paragraphe, on appellera

- **modèle complet** le modèle avec toutes les variables explicatives, c'est à dire le modèle de régression linéaire $Y = X\beta + W$.
- **modèle restreint** le modèle valide sous H_0, c'est à dire le modèle

  $$Y = X_r\beta_r + W$$

  avec

  $$X_r = \begin{pmatrix} 1 & x_{1,q+1} & \dots & x_{1,p} \\ 1 & x_{2,q+1} & \dots & x_{2,p} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n,q+1} & \dots & x_{n,p} \end{pmatrix}$$

  et $\beta_r = (\beta_0, \beta_{q+1}, ..., \beta_p)'$.

Lorsqu'on ajuste le modèle complet, on rappelle que $\hat{Y} = X(X'X)^{-1}X'Y = X\hat{B}$, avec $\hat{B}$ l'estimateur des moindres carrés de β, est la projection orthogonale de Y sur $E = Im(X) = \{X\beta | \beta \in \mathbb{R}^{p+1}\}$. Pour le modèle restreint, nous noterons de même $\hat{Y}_r = X_r(X_r'X_r)^{-1}X_r'Y$ le projeté orthogonal de Y sur $E_r = Im(X_r)$. E_r est un sev de E de dimension $p + 1 - q$. Notons de même $\hat{Y}_r$ la projection orthogonale de Y sur E_r. Le test d'analyse de la variance est basé sur la proposition suivante.

Proposition 2 *Sous les hypothèses du modèle linéaire gaussien, si l'hypothèse* $H_0 : \beta_1 = 0, \beta_2 = 0, ..., \beta_q = 0$ *est vraie, alors*

$$F_c = \frac{n-p-1}{q} \frac{\left\|\hat{Y} - \hat{Y}_r\right\|^2}{\left\|Y - \hat{Y}\right\|^2} \sim \mathcal{F}_{q,n-p-1}$$

avec $\mathcal{F}_{q,n-p-1}$ *la loi de Fisher de degrés de liberté* q *et* $n-p-1$.

Preuve 1 *On a*

$$W = Y - X\beta = (Y - \hat{Y}) + (\hat{Y} - \hat{Y}_r) + (\hat{Y}_r - X\beta)$$

avec

- $(Y - \hat{Y}) = \pi_{E^\perp}(Y) \in E^\perp$,
- $(\hat{Y} - \hat{Y}_r) \in E$ *car* $\hat{Y} \in E$ *et* $\hat{Y}_r \in E$,
- $(\hat{Y} - \hat{Y}_r) \in E_r^\perp$ *car* $\hat{Y}_r = \pi_{E_r}(Y) = \pi_{E_r}(\pi_E(Y))$. *On a donc* $\hat{Y}_r = \pi_{E_r}(\hat{Y})$ *car* $E_r \subset E$,
- $(\hat{Y}_r - X\beta) \in E_r$ *si* H_0 *est vraie, car on a alors* $X\beta \in E_r$.

On en déduit en particulier que $\hat{Y} - \hat{Y}_r = \pi_{E \cap E_r^\perp}(W)$. *De plus, on a*

$$\mathbb{R}^n = E \oplus^\perp E^\perp = E_r \oplus^\perp (E \cap E_r^\perp) \oplus^\perp E^\perp$$

et donc $dim(E \cap E_r^\perp) = n - dim(E_r) - dim(E^\perp) = q$. *D'après le théorème de Cochran, on a alors*

- $\hat{Y}_r - \hat{Y}$ *et* $Y - \hat{Y}$ *indépendants et donc* $\left\|\hat{Y} - \hat{Y}_r\right\|^2$ *et* $\left\|Y - \hat{Y}\right\|^2$ *indépendants,*
- $\frac{1}{\sigma^2}\left\|\hat{Y} - \hat{Y}_r\right\|^2 \sim \chi^2_{dim(E \cap E_r^\perp)}$ *avec* $dim(E \cap E_r^\perp) = q$,
- $\frac{1}{\sigma^2}\left\|Y - \hat{Y}\right\|^2 \sim \chi^2_{dim(E^\perp)}$ *avec* $dim(E^\perp) = n - p - 1$

On déduit aisément le résultat.

Application à la construction du test de Fisher (ou test d'analyse de la variance): on accepte H_0 si et seulement si $F \leq f_{1-\alpha,q,n-p-1}$. En pratique, on accepte H_0 lorsque $\left\|\hat{Y} - \hat{Y}_r\right\|^2$ est suffisamment petit, c'est à dire si les projetés orthogonaux de Y sur E et E_r sont proches (intuitivement, cela signifie qu'on perd peu d'information en projetant sur E_r au lieu de projeter sur E). La p-value du test est

$$p_v = P(F > f)$$

avec f la valeur prise par la statistique de test et $F \sim F_{1-\alpha,q,n-p-1}$.

Illustration avec R. En pratique, pour réaliser le test d'analyse de la variance avec R, il faut ajuster le modèle complet et le modèle restreint avec la fonction *lm* puis utiliser la fonction *anova* pour réaliser le test.

```
fit=lm(Sepal.Length~.,data=iris[,-5])  #ajustement modèle complet
fit0=lm(Sepal.Length~Petal.Length,data=iris)  #ajustement modèle réduit
anova(fit0,fit)  #réalisation du test
```

```
## Analysis of Variance Table
##
## Model 1: Sepal.Length ~ Petal.Length
## Model 2: Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    148 24.525
## 2    146 14.445  2     10.08 50.938 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# La p_value du test de l'hypothèse
# H0 : beta_1=\beta_3=0
# est très faible. H_0 est refusée.
# On en déduit que le modèle complet est meilleur que le modèle restreint
```

Cas particulier 1. Si $q = 1$, on teste l'hypothèse $H_0 : \beta_1 = 0$. On peut montrer qu'on retrouve le test de Student du cours de L3.

Illustration avec R

```
summary(fit)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ ., data = iris[, -5])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.82816 -0.21989  0.01875  0.19709  0.84570
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.85600    0.25078   7.401 9.85e-12 ***
## Sepal.Width   0.65084    0.06665   9.765  < 2e-16 ***
## Petal.Length  0.70913    0.05672  12.502  < 2e-16 ***
## Petal.Width  -0.55648    0.12755  -4.363 2.41e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3145 on 146 degrees of freedom
## Multiple R-squared:  0.8586, Adjusted R-squared:  0.8557
## F-statistic: 295.5 on 3 and 146 DF,  p-value: < 2.2e-16
```

```
#la variable Petal.Width a un effet significatif (p-value 2.41e-05)
#On va retrouver la p-value du test avec le test de Fisher
fit1=lm(Sepal.Length~Sepal.Width+Petal.Length,data=iris)
#ajustement modèle réduit, ici modèle à 2 variables sans Petal.Width
anova(fit1,fit)  #réalisation du test
```

```
## Analysis of Variance Table
##
## Model 1: Sepal.Length ~ Sepal.Width + Petal.Length
## Model 2: Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    147 16.329
## 2    146 14.445  1    1.8834 19.035 2.413e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#les p-values des tests de Student et de Fisher
#sont identiques (2.41e-05)
```

Cas particulier 2. Si $q = p$, on teste l'hypothèse $H_0 : \beta_1 = 0, \beta_2 = 0, ..., \beta_p = 0$. L'hypothèse H_0 signifie qu'aucune variable explicative n'a d'effet sur la réponse. Sous H_0, on a $Y_i = \beta_0 + W_i$, c'est à dire

$$(Y_1, ..., Y_n) \sim_{iid} \mathcal{N}(\beta_0, \sigma^2)$$

Sous H_0, les observations proviennent d'un échantillon i.i.d d'une loi normale d'espérance β_0 et de variance σ^2. On a alors $X_r = e$ avec $e = (1, ..., 1)'$, $\hat{B}_r = \bar{Y}$, $\hat{Y}_r = \bar{Y}e$ ($\bar{Y}$ est la meilleur prédiction de Y_i au sens des moindres carrés lorsqu'on ne dispose d'aucune variable explicative). La statistique de test s'écrit donc

$$F_c = \frac{n-p-1}{p} \frac{\left\|\hat{Y} - \bar{Y}e\right\|^2}{\left\|Y - \hat{Y}\right\|^2} = \frac{n-p-1}{p} \frac{\sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2}{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}.$$

Si $H_0 : \beta_1 = 0, \beta_2 = 0, ..., \beta_p = 0$ est vraie, alors $F_c \sim \mathcal{F}_{p,n-p-1}$. Notons alors

- $SC_{tot} = \left\|Y - \bar{Y}e\right\|^2 = \sum_{i=1}^n (Y_i - \bar{Y})^2$ la **somme des carrés totale**.
- $SC_{reg} = \left\|\hat{Y} - \bar{Y}e\right\|^2 = \sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2$ la **somme des carrés expliquée par la régression**.
- $SC_{res} = \left\|Y - \hat{Y}\right\|^2 = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$ la **somme des carrés résiduelle**.

On a alors $F_c = \frac{n-p-1}{p} \frac{SC_{reg}}{SC_{res}}$. Afin de mesurer la qualité du modèle, on utilise généralement le **coefficient de détermination** (ou "**coefficient de corrélation multiple**")

$$R^2 = \frac{SC_{reg}}{SC_{tot}}.$$

Par ailleurs, d'après le théorème de Pythagore, on a la formule d'analyse de la variance

$$SC_{tot} = SC_{res} + SC_{reg}$$

et donc

$$F_c = \frac{n-p-1}{p} \frac{R^2}{1-R^2}.$$

On accepte $H_0 : \beta_1 = 0, \beta_2 = 0, ..., \beta_p = 0$ si et seulement si

$$F_c \leq f_{1-\alpha,p,n-p-1} \iff R^2 \leq \frac{1}{1 + \frac{n-p-1}{p f_{1-\alpha,p,n-p-1}}}.$$

On rappelle que R^2 représente la proportion de **variation totale** expliquée par le modèle et vérifie $R^2 = cos(\theta)^2$, avec θ l'angle entre les vecteurs $Y - \bar{Y}e$ et $\hat{Y} - \bar{Y}e$. Si R^2 est proche de 0, alors les variables explicatives apportent peu d'information sur la réponse ($\hat{Y} \approx \bar{Y}$) et la statistique de test F_c prendra aussi des valeurs proches de 0.

Illustration avec R. La fonction *summary* associée à la fonction *lm* donne le résultat du test d'analyse de la variance dans le cas particulier $p = q$.

```
summary(fit)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ ., data = iris[, -5])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.82816 -0.21989  0.01875  0.19709  0.84570
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.85600    0.25078   7.401 9.85e-12 ***
## Sepal.Width   0.65084    0.06665   9.765  < 2e-16 ***
## Petal.Length  0.70913    0.05672  12.502  < 2e-16 ***
```

```
## Petal.Width  -0.55648    0.12755  -4.363 2.41e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3145 on 146 degrees of freedom
## Multiple R-squared:  0.8586, Adjusted R-squared:  0.8557
## F-statistic: 295.5 on 3 and 146 DF,  p-value: < 2.2e-16
```

```
#La dernière ligne du summary donne le résultat du test de Fisher associé à l'hypothèse
#H0 : beta1=beta2=beta3=0
#Ici la p-value très faible, H_0 est refusée
#Interprétation : les variables explicatives apportent de l'information sur la réponse
#On peut retrouver le résultat avec la fonction anova
fit2=lm(Sepal.Length~1,data=iris)  #ajustement modèle avec seulement intercept
anova(fit2,fit)  #on retrouve la même p-value que avec summary
```

```
## Analysis of Variance Table
##
## Model 1: Sepal.Length ~ 1
## Model 2: Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1    149 102.168
## 2    146  14.445  3    87.723 295.54 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Cas particulier 3. Dans le cas particulier où $p = q = 1$ (régression linéaire simple), les tests de Fisher et de Student sont donc équivalents et la fonction *summary* renvoie deux fois la même p-value (cf code ci-dessous). Par ailleurs, on peut montrer que $R^2 = r^2$ avec r le coefficient de corrélation usuel. Le test peut alors s'interpréter comme un test de significativité de la corrélation entre deux variables (sous les hypothèses du modèle linéaire gaussien), et on peut également retrouver la p-value du test avec la fonction *cor.test*.

```
fit3=lm(Sepal.Length~Sepal.Width,data=iris)  #modèle avec p=q=1
summary(fit3)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Sepal.Width, data = iris)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5561 -0.6333 -0.1120  0.5579  2.2226
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.5262     0.4789   13.63   <2e-16 ***
## Sepal.Width  -0.2234     0.1551   -1.44    0.152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8251 on 148 degrees of freedom
## Multiple R-squared:  0.01382,    Adjusted R-squared:  0.007159
## F-statistic: 2.074 on 1 and 148 DF,  p-value: 0.1519
```

```r
#les p-values des tests de Student et de Fisher  de l'hypothèse
#H0 : beta_1=0
#sont identiques (ici 0.1519).
#Sepal.Width n'a pas d'effet significatif sur Sepal.Length
cor.test(iris$Sepal.Length,iris$Sepal.Width)
```

```
##
##  Pearson's product-moment correlation
##
## data:  iris$Sepal.Length and iris$Sepal.Width
## t = -1.4403, df = 148, p-value = 0.1519
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.27269325  0.04351158
## sample estimates:
##        cor
## -0.1175698
```

```r
# On retrouve encore la même p-value (0.152)
# Sur cet exemple, la corrélation entre les deux variables n'est pas significative
```

3.2 Sélection de variables

3.2.1 Introduction

L'objectif des méthodes de sélection de variables en régression est d'éliminer les variables explicatives qui apportent trop peu d'information sur la variable à expliquer. Pourquoi est-il important de faire de la sélection de modèle?

- Ceci permet d'obtenir un modèle plus facilement **interprétable**, sans variables 'inutiles' qui n'ont pas d'effet sur la variable à expliquer (cf exercice 1 du TD).
- Ceci permet de réduire le nombre de paramètres inconnus à estimer et donc de **réduire l'incertitude** dans le modèle. L'ajustement d'un modèle avec des variables explicatives "inutiles" conduit généralement à des prévisions avec une plus grande variance que celles obtenus avec un modèle contenant seulement les variables "utiles" (cf exercice 2). La sélection de modèle permet de trouver un bon compromis entre la complexité du modèle ajusté (c'est à dire le nombre de paramètres inconnus) et le nombre d'observations disponibles n. Lorsque le modèle est trop complexe, il pourra expliquer très bien les données sur lesquelles il a été ajusté (par exemple si $p+1=n$ et $rang(X)=n$ alors on peut écrire y comme une combinaison linéaire des variables explicatives et le modèle expliquera parfaitement la variable à expliquer sur l'ensemble d'apprentissage comme dans l'exemple R ci-dessous) mais se généralisera très mal sur des nouveaux individus qui n'ont pas été utilisés pour ajuster le modèle. On parle alors de '**surapprentissage**' ('overfitting').
- Ceci permet de réduire les temps de calculs.

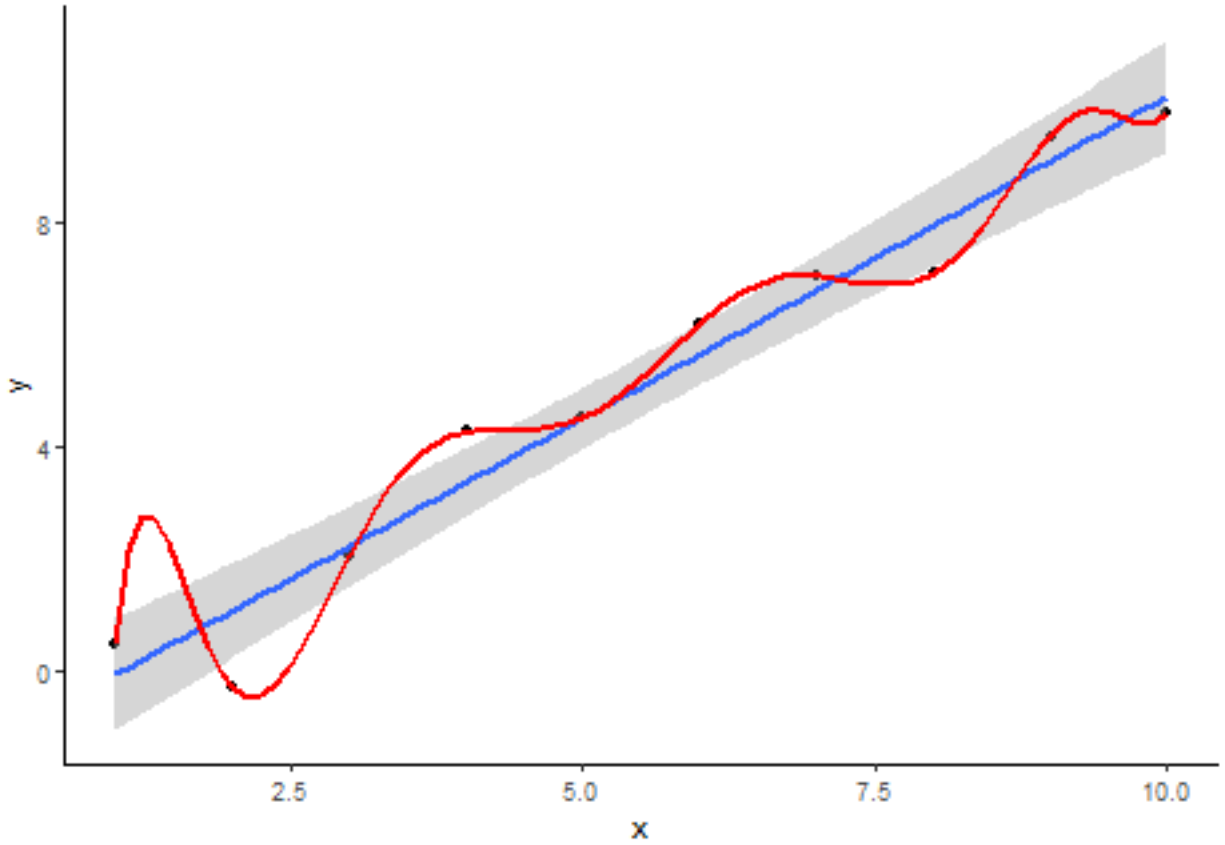
```r
#Un exemple de régression polynomiale avec surapprentissage
set.seed(231)
n=10  #taille échantillon simulé
x=1:n #variable explicative
y=x+rnorm(n)  #simulation de y selon un modèle de régression linéaire simple
z=data.frame(x=x,y=y) #création d'un data.frame avec les données
p=9

library(ggplot2)
ggplot(z,aes(y=y,x=x))+
  geom_point()+
```

```
  geom_smooth(method = lm,formula = y ~x,interval='prediction')+
  geom_smooth(method = lm,formula = y ~  poly(x,p,raw='TRUE'),col='red')+
  theme_classic()
```

```
## Warning in geom_smooth(method = lm, formula = y ~ x, interval = "prediction"):
## Ignoring unknown parameters: ‘interval‘
```

```
## Warning in qt((1 - level)/2, df): Production de NaN
```

```
## Warning in max(ids, na.rm = TRUE): aucun argument pour max ; -Inf est renvoyé
```




```
#Le polynôme de degré 9 passe exactement par les observations (courbe rouge)
#Le modèle est 'parfait' sur l'ensemble d'apprentissage
#Problème : donne de mauvaises prévisions en dehors de l'ensemble d'apprentissage
#Il vaut mieux utiliser le modèle de régression linéaire simple pour faire de la prédiction (droite bleue)

fit=lm(y~poly(x,p,raw='TRUE'))  #régression polynomiale, polynôme de degré p
#autant de covariables que d'individus (n=p+1)!
summary(fit)  #on remarque que R^2=1
```

```
##
## Call:
## lm(formula = y ~ poly(x, p, raw = "TRUE"))
##
## Residuals:
## ALL 10 residuals are 0: no residual degrees of freedom!
##
```

```
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -1.832e+02        NaN     NaN      NaN
## poly(x, p, raw = "TRUE")1   5.074e+02        NaN     NaN      NaN
## poly(x, p, raw = "TRUE")2 -5.563e+02        NaN     NaN      NaN
## poly(x, p, raw = "TRUE")3   3.222e+02        NaN     NaN      NaN
## poly(x, p, raw = "TRUE")4 -1.103e+02        NaN     NaN      NaN
## poly(x, p, raw = "TRUE")5   2.343e+01        NaN     NaN      NaN
## poly(x, p, raw = "TRUE")6 -3.124e+00        NaN     NaN      NaN
## poly(x, p, raw = "TRUE")7   2.543e-01        NaN     NaN      NaN
## poly(x, p, raw = "TRUE")8 -1.154e-02        NaN     NaN      NaN
## poly(x, p, raw = "TRUE")9   2.237e-04        NaN     NaN      NaN
##
## Residual standard error: NaN on 0 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:    NaN
## F-statistic:   NaN on 9 and 0 DF,  p-value: NA
```

```
#Rq : la variance empirique du résidu est égale à 0 et la matrice X'X n'est pas inversible
#ceci pose des problèmes numériques (NaN dans le summary)
```

La sélection de modèle est particulièrement difficile lorsque les variables explicatives sont (fortement) corrélées entre elles. En effet, il est alors difficile de choisir les variables les plus pertinentes puisqu'elles apportent une information similaire sur la variable explicative. Pour les applications pratiques en actuariat, les variables explicatives fortement corrélées entre elles sont parfois enlevées 'à la main' avant d'ajuster le modèle. Par exemple si deux variables sont fortement corrélées entre elles, on en conserve seulement une des deux.

Une autre pratique courante pour les applications en actuariat consiste à enlever les variables explicatives qui ne sont pas corrélées avec la variable à expliquer avant d'ajuster un modèle de régression. Ceci n'est pas toujours pertinent, même dans le cas du modèle linéaire (cf exercice 3 du TD).

A retenir : il n'est pas nécessaire d'enlever au préalable certaines variables explicatives d'un jeu de données (par exemple les variables non-corrélées avec la réponse ou fortement corrélées entre elles). Les méthodes discutées dans ce paragraphe permettent de faire la sélection de variables de manière automatique et optimale!

3.2.2 Méthodes exhaustives

Le principe est simple : on choisit un critère qui permet de mesurer la qualité d'un modèle, on calcule ce critère pour tous les 2^p modèles réduits possibles (i.e. les modèles obtenus en faisant l'hypothèse que certains coefficients sont égaux à 0) et on sélectionne le "meilleur modèle". Différents critères existent dans la littérature pour mesurer la qualité d'un modèle. Les plus usuels sont définis ci-dessous.

- **Coefficient** $R^2 = \frac{SC_{reg}}{SC_{tot}}$ qui mesure la proportion de variation totale expliquée par le modèle. **Problème :** si le modèle 1 est un sous modèle du modèle 2, alors $R_2^2 \geq R_1^2$ avec R_i^2 le coefficient R^2 du modèle i (car si $E_1 \subset E_2$ alors $\|Y - \pi_{E_1}(Y)\| \geq \|Y - \pi_{E_2}(Y)\|$) et ce critère sélectionne donc systématiquement le modèle complet.
- **Coefficient du R^2 ajusté** : on peut réécrire $R^2 = 1 - \frac{SC_{res}}{SC_{tot}} = 1 - \frac{SC_{res}/n}{SC_{tot}/n}$ avec
  - $\frac{SC_{res}}{n} = \frac{1}{n}\sum(Y_i - \hat{Y}_i)^2$ un estimateur biaisé de σ^2 la variance du résidu du modèle linéaire,
  - $\frac{SC_{tot}}{n} = \frac{1}{n}\sum(Y_i - \bar{Y})^2$ un estimateur biaisé de $var(Y)$ (sous l'hypothèse que l'échantillon est i.i.d.).

  On remplace alors ces estimateurs par les estimateurs non biaisés, et on définit le **coefficient du R^2 ajusté** par $R^2_{aj} = 1 - \frac{SC_{res}/(n-p-1)}{SC_{tot}/(n-1)} = \frac{(n-1)R^2-p}{n-p-1}$. R^2_{aj} s'interprète comme la proportion de **variance** expliquée par le modèle. Ce critère est donné par la fonction summary.lm de R.
- **Critère de log-vraisemblance pénalisée**. Il semble naturel de sélectionner le modèle le plus "vraisemblable", la vraisemblance d'un modèle étant définie par $L(\hat{\theta}) = p(y_1, ..., y_n|\hat{\theta})$ où L désigne la fonction de vraisemblance et $\hat{\theta}$ l'estimateur du maximum de vraisemblance. A nouveau, si le modèle 1 est un sous modèle du modèle

2 alors la vraisemblance du modèle 2 est toujours plus grande que celle du modèle 1 (car si $\Theta_1 \subset \Theta_2$ alors $\sup_{\theta \in \Theta_1} L(\theta) < \sup_{\theta \in \Theta_2} L(\theta)$). On "pénalise" alors la vraisemblance par le nombre de paramètres inconnus. Deux critères sont couramment utilisés :

- $AIC = -2log(L(\hat{\theta})) + 2n_{par}$ avec n_{par} le nombre de paramètres inconnus,
- $BIC = -2log(L(\hat{\theta})) + log(n) * n_{par}$ avec n le nombre d'individus.

On choisit ensuite le modèle pour lequel la valeur du AIC ou BIC est la plus faible, ce qui permet de trouver un compromis entre un modèle avec une grande vraisemblance mais un nombre de paramètres restreint. Lorsque $log(n) > 2$, la pénalisation associée à BIC est plus importante que celle associée à AIC : BIC conduit donc à sélectionner des modèles avec moins de paramètres (ou "plus parcimonieux"). Dans le cas particulier du modèle linéaire gaussien, on peut vérifier que

$$log(L(\hat{\theta})) = -nlog(\hat{\sigma}) - \frac{n}{2}\ln(2\pi) - \frac{n}{2}$$

avec $\hat{\sigma}^2 = \frac{1}{n}\sum(Y_i - \hat{Y}_i)^2$ l'erreur quadratique moyenne de l'erreur de prédiction qui est un estimateur biaisé de σ^2. Les critères AIC et BIC cherchent donc un compromis entre entre un modèle avec une faible erreur de prédiction (sur l'ensemble d'apprentissage) tout en ayant un nombre de paramètres restreint.

- **Mesure de la qualité prédictive d'un modèle par validation croisée.** Pour de nombreuses applications, on cherche à obtenir le modèle qui donne les meilleurs prévisions. Afin de mesurer la qualité prédictive d'un modèle, on utilise alors la méthode **validation croisée**. L'idée de base est de diviser le jeu de données en 2 sous-échantillons.
  - Le premier échantillon (appelé **ensemble d'apprentissage**) est utilisé pour ajuster le modèle (c'est à dire estimer les paramètres).
  - Le deuxième échantillon (appelé **ensemble de test** ou **ensemble de validation**) est utilisé pour calculer une erreur de prévision (score). Le critère le plus usuel pour les variables quantitatives est l'erreur quadratique moyenne (Root Mean Square Error) sur l'ensemble de test défini par

    $$RMSE = \sqrt{\frac{1}{\#test}\sum_{i \in test}(y_i - \hat{y}_i)^2}$$

    avec $\hat{y}_i$ la valeur prédite par le modèle ajusté sur l'ensemble d'apprentissage et y_i la valeur observée. Selon l'application, on pourra considérer d'autres critères, comme par exemple pour les variables quantitatives
    - Mean Absolute Error $MAE = \frac{1}{\#test}\sum_{i \in test}|y_i - \hat{y}_i|$
    - Mean absolute percentage error $MAPE = 100 * \frac{1}{\#test}\sum_{i \in test}|\frac{y_i - \hat{y}_i}{y_i}|$

    Il existe aussi des mesures de performance spécifiques pour les variables binaires, cf par exemple https://en.wikipedia.org/wiki/Receiver_operating_characteristic

On vérifie aisément que si l'ensemble d'apprentissage et l'ensemble de validation sont les mêmes, alors le modèle complet sera toujours meilleurs en terme de RMSE car le critère choisi est directement lié à la fonction utilisée dans la méthode des moindres carrés. Choisir des ensembles distincts pour l'apprentissage et la validation permet d'éviter le sur-apprentissage en vérifiant la capacité du modèle à se généraliser à des individus qui n'ont pas été utilisés dans la phase d'ajustement. Plusieurs méthodes peuvent être utilisées pour construire les ensembles d'apprentissage et de test.

- **'Holdout' ou 'testset' cross-validation**.
  - L'ensemble d'apprentissage est obtenu en choisissant aléatoirement un certain pourcentage des individus (usuellement 60%).
  - L'ensemble de test est constitué des individus restants.

  Cette méthode nécessite un grand nombre d'individus n, sinon les résultats obtenus vont être sensibles au choix des ensembles d'apprentissage et de validation.
- **'k-fold' cross-validation**. On commence par diviser de manière aléatoire les observations en k blocs distincts de même taille. Ensuite, pour chacun des blocs $i \in \{1, ...k\}$,
  1. on ajuste le modèle en utilisant comme ensemble d'apprentissage les $k-1$ blocs restants,

2. on calcule la RMSE (ou un autre score) sur l'ensemble de validation constitué du bloc i, noté $RMSE(i)$.

Finalement la qualité du modèle est mesuré par la RMSE moyenne sur les k blocs c'est à dire

$$RMSE = \sqrt{\frac{1}{k}\sum_{i=1}^{k} RMSE(i)^2}$$

– **"Leave-one out cross-validation"**. Cette méthode est un cas particulier de la méthode k-fold cross-validation lorsque $k = n$. Pour les petits échantillons, ceci permet de créer un ensemble d'apprentissage de taille $n - 1$ pour ajuster le modèle et donc de réduire au minimum la taille de l'échantillon utilisé dans la phase d'apprentissage.

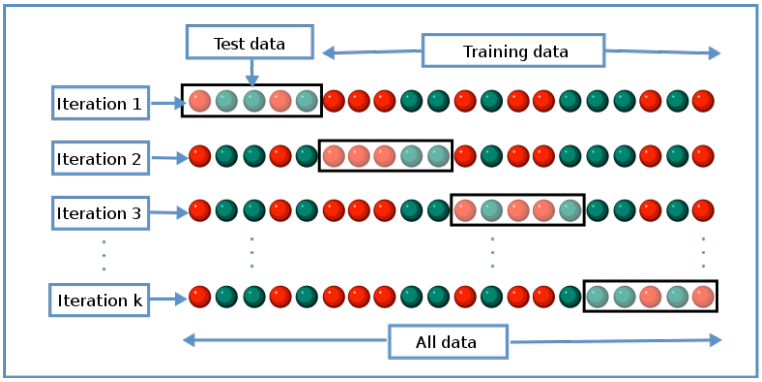


Figure 1: Schéma illustrant la méthode de validation croisée. Source : https://en.wikipedia.org/wiki/Cross-validation_(statistics).

Remarque 1 *Les critères du R^2 et du R^2_{aj} sont spécifiques au modèle linéaire. Les critères de log-vraisemblance pénalisée sont plus généraux et sont couramment utilisés pour de nombreux* ***modèles paramétriques*** *pour lesquels on peut écrire une fonction de vraisemblance (ajustement de lois, modèles linéaires et GLM, séries temporelles,...). Ceux basés sur le calcul d'une erreur de prévision par validation croisée ne nécessitent pas de formuler un modèle paramétrique et sont les plus couramment utilisés en apprentissage statistique (par exemple pour les forêts aléatoires ou les réseaux de neurones). La mise en oeuvre de la validation croisée peut nécessiter des temps de calcul importants (selon la taille des jeux de données et la complexité des modèles utilisés).*

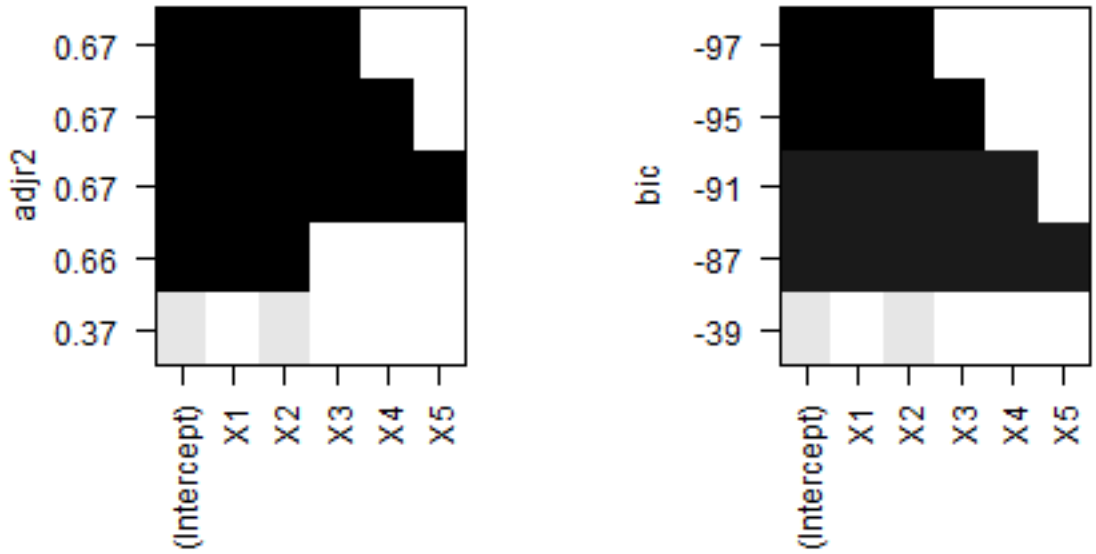
Selection de variable exhaustive avec R

Plusieurs packages sont disponibles dans R pour faire de la sélection de modèle avec les méthodes exhaustives. On pourra par exemple utiliser les packages *leaps* (plus complet mais utilisable uniquement dans le cadre du modèle linéaire) ou *bestglm* (permet de faire également la sélection de modèle pour les GLM).

On pourra consulter la vignette du package bestglm pour plus de détails http://cran.nexr.com/web/packages/bestglm/vignettes/bestglm.pdf

```
#Illustration avec l'exemple de l'exercice 2 du TD
##########
#Simulation des données
#5 variables explicatives indépendantes
# Seulement les deux premières variables ont un effet sur Y
n=10^2  #nombre d'individus
p=5    #nombre de variables explicatives
X=cbind(rep(1,n),matrix(rnorm(n*p),ncol=p))  #simulation variables explicatives
beta=c(rep(1,3),rep(0,p-2))  #paramètres
y=X%*%beta+rnorm(n)  #simulation de la variable à expliquer
data=data.frame(X[,-1],y=y)
##################
```

```r
#Sélection de variable avec regsubsets
library(leaps)  #il existe d'autres packages
a=regsubsets(y~.,nvmax=p,data=data,method='exhaustive') #V1 est la variable à expliquer
#nvmax : nombre de variables maximal
#représentation graphique
par(mfrow=c(1,2))
plot(a,scale="adjr2")
#Montre les meilleurs modèles à 1, 2, ..., nvmax=5 variables
#Les modèles sont triés en fonction du R^2 ajustés
plot(a,scale="bic") #Idem avec BIC
```




```r
#En général, le 'vrai' modèle qui a servi à simuler les données est identifié
##################
#Sélection de variable avec bestglm
library(bestglm)
```

```
## Warning: le package 'bestglm' a été compilé avec la version R 4.2.3
```

```r
sel= bestglm(data,IC="BIC")  #avec critère BIC
#NB. La variable à expliquer doit être mise en dernière position dans le data.frame
sel
```

```
## BIC
## BICq equivalent for q in (8.62065974160942e-12, 0.741869610858623)
## Best Model:
##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 1.1054120 0.09982640 11.073343 6.577050e-19
## X1          0.9258723 0.10021798  9.238585 5.912388e-15
## X2          1.1250685 0.09466507 11.884727 1.232020e-20
```

```
sel$Subsets
```

```
##    (Intercept)    X1    X2    X3    X4    X5 logLikelihood        BIC
## 0         TRUE FALSE FALSE FALSE FALSE FALSE    -52.981348 105.962695
## 1         TRUE FALSE  TRUE FALSE FALSE FALSE    -28.983718  62.572606
## 2*        TRUE  TRUE  TRUE FALSE FALSE FALSE      2.577527   4.055287
## 3         TRUE  TRUE  TRUE  TRUE FALSE FALSE      3.824403   6.166704
## 4         TRUE  TRUE  TRUE  TRUE  TRUE FALSE      4.183301  10.054079
## 5         TRUE  TRUE  TRUE  TRUE  TRUE  TRUE      4.440356  14.145140
```

```
#meilleurs modèles à 1, 2, ... variables
sel= bestglm(data,IC="LOOCV")  #avec Leave One Out Cross-Validation
sel$Subsets
```

```
##    (Intercept)    X1    X2    X3    X4    X5 logLikelihood    LOOCV
## 0         TRUE FALSE FALSE FALSE FALSE FALSE    -52.981348 2.943878
## 1         TRUE FALSE  TRUE FALSE FALSE FALSE    -28.983718 1.865235
## 2         TRUE  TRUE  TRUE FALSE FALSE FALSE      2.577527 1.017284
## 3*        TRUE  TRUE  TRUE  TRUE FALSE FALSE      3.824403 1.009392
## 4         TRUE  TRUE  TRUE  TRUE  TRUE FALSE      4.183301 1.025446
## 5         TRUE  TRUE  TRUE  TRUE  TRUE  TRUE      4.440356 1.045375
```

3.2.3 Méthodes pas à pas

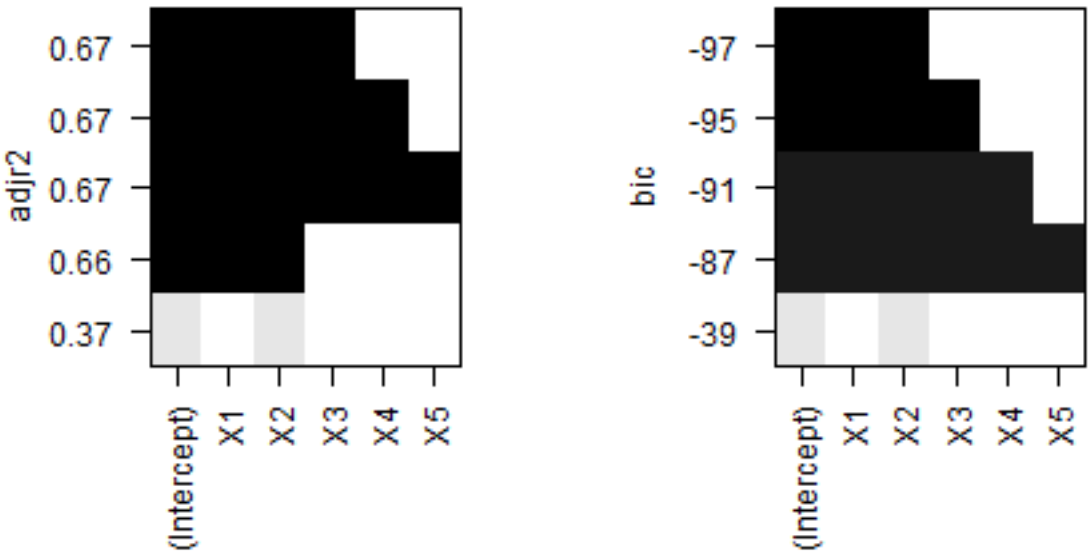
Lorsque le nombre p de variables est important, il n'est plus possible d'étudier tous les 2^p sous modèles possibles. Sur l'exemple R précédent, les temps de calculs deviennent conséquents lorsque $p = 40$ (cf exemple R à la fin du paragraphe). On peut alors utiliser une méthode pas à pas afin d'éviter une recherche exhaustive. Ces méthodes permettent généralement de trouver un "bon" modèle mais pas forcément le "meilleur" modèle.

- **Elimination en arrière (backward)**. L'algorithme démarre du modèle complet. À chaque étape, la variable la 'moins utile' est supprimée. L'algorithme s'arrête lorsque le modèle ne s'améliore plus lorsqu'on enlève une variable. Différents critères peuvent être utilisés pour choisir la variable à éventuellement supprimer à une étape donnée de l'algorithme.
  - Utilisation de la p-value du test de Student de l'hypothèse $H_i : \beta_i = 0$. On retire alors la variable avec la plus grand p-value et on s'arrête lorsque toutes les p-values sont inférieures à 5%.
  - La fonction *stepAIC* de *R* utilise le critère AIC.
- **Sélection en avant (forward)** On fait la même chose en partant du modèle réduit avec tous les coefficients nuls (sauf β_0) et on ajoute successivement les variables les plus 'utiles' une à une. La méthode d'élimination en avant est souvent plus économique puisqu'elle permet d'éviter de travailler avec plus de variables que nécessaire. Un inconvénient est qu'une fois qu'une variable est introduite, elle ne peut plus être éliminée et qu'on obtient donc parfois des modèles avec des variables non significatives. Ce problème est résolu avec la méthode stepwise.
- **Méthode mixte (stepwise)** C'est la méthode la plus couramment utilisée dans les logiciels. A chaque itération de la méthode de sélection en avant, on effectue une étape d'élimination en arrière: on vérifie si toutes les variables sont 'utiles', et le cas échéant on enlève celles qui ne le sont pas, successivement comme dans la méthode d'élimination en arrière. On peut aussi procéder dans l'autre sens et utiliser un algorithme backward à chaque itération en intercalant une étape forward (c'est la méthode par défaut utilisée par la fonction *stepAIC*).

Selection de modèle pas à pas avec regsubsets (uniquement pour le modèle linéaire)

```
a=regsubsets(y~.,nvmax=p,data=data,method='seqrep')
#nvmax : nombre de variables maximal
#method='seqrep' : stepwise selection
```

```
#recherche exhaustive avec R2_aj
par(mfrow=c(1,2))
plot(a,scale="adjr2")
plot(a,scale="bic")
```



Selection de modèle pas à pas avec stepAIC (peut aussi être utilisé pour les GLM)

```
library(MASS)
#méthode backward
fit=lm(y~.,data=data)  #modèle initial
#fit2=stepAIC(fit,direction="backward")
#méthode forward
#fit0=lm(y~1,data=data)  #modèle initial
#fit3=stepAIC(fit0,scope=list(lower=~1,upper=~X1+X2+X3+X4+X5),direction="forward")
#scope décrit le modèle le plus simple/complexe à ajuster
#summary(fit3)
fit3=stepAIC(fit,scope=list(lower=~1,upper=~.),direction="both")
```

```
## Start:  AIC=3.12
## y ~ X1 + X2 + X3 + X4 + X5
##
##        Df Sum of Sq     RSS    AIC
## - X5    1     0.472  91.974  1.633
## - X4    1     0.683  92.186  1.863
## <none>               91.502  3.119
## - X3    1     2.363  93.865  3.669
## - X1    1    81.717 173.219 64.939
## - X2    1   136.304 227.806 92.333
##
```

```
## Step:  AIC=1.63
## y ~ X1 + X2 + X3 + X4
##
##        Df Sum of Sq     RSS    AIC
## - X4    1     0.663  92.636  0.351
## <none>               91.974  1.633
## - X3    1     2.330  94.303  2.135
## + X5    1     0.472  91.502  3.119
## - X1    1    81.309 173.282 62.975
## - X2    1   138.796 230.770 91.625
##
## Step:  AIC=0.35
## y ~ X1 + X2 + X3
##
##        Df Sum of Sq     RSS    AIC
## <none>               92.636  0.351
## - X3    1     2.339  94.976  0.845
## + X4    1     0.663  91.974  1.633
## + X5    1     0.451  92.186  1.863
## - X1    1    80.677 173.314 60.993
## - X2    1   139.615 232.251 90.265
```

```
summary(fit3)
```

```
##
## Call:
## lm(formula = y ~ X1 + X2 + X3, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5505 -0.6270  0.1134  0.6008  2.3056
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.09775    0.09922  11.063  < 2e-16 ***
## X1           0.91289    0.09984   9.144 1.03e-14 ***
## X2           1.13150    0.09407  12.028  < 2e-16 ***
## X3          -0.16997    0.10917  -1.557    0.123
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9823 on 96 degrees of freedom
## Multiple R-squared:  0.6789, Adjusted R-squared:  0.6689
## F-statistic: 67.67 on 3 and 96 DF,  p-value: < 2.2e-16
```

```
#algo stepwise avec l'algo backward intercallé dans l'algo forward (cf cours)
```

Comparaison des temps de calcul lorsque le nombre de covariables augmente

```
n=100;p=35   #tester avec p=40 ou p=45
X=cbind(rep(1,n),matrix(rnorm(n*p),ncol=p))  #covariables
beta=c(rep(1,3),rep(0,p-2))  #coefficients
y=X%*%beta+rnorm(n)  #simulation de la réponse
data=as.data.frame(cbind(y,X[,-1]))
system.time(regsubsets(V1~.,nvmax=p,data=data,method='exhaustive'))
```

```
## utilisateur     système      écoulé
##        1.91        0.00        2.14
```

```
system.time(regsubsets(V1~.,nvmax=p,data=data,method='seqrep'))
```

```
## utilisateur     système      écoulé
##        0.02        0.00        0.02
```

```
fit=lm(V1~.,data=data)
system.time(stepAIC(fit,trace=0))
```

```
## utilisateur     système      écoulé
##        0.14        0.00        0.16
```

```
#trace=0 enlève les affichages
#Les méthodes pas à pas sont plus rapides
```

3.3 Régression régularisée ou pénalisée

3.3.1 Régression Ridge

On peut vérifier que la matrice $X'X$ n'est pas inversible lorsque $p > n$ ou lorsque les covariables ne sont pas linéairement indépendantes, et dans ces cas l'estimateur des moindres carrés $\hat{B} = (X'X)^{-1}X'Y$ n'est donc pas défini.

Dans le cas où la matrice $X'X$ est inversible la variance des estimateurs des moindres carrés (et donc des prévisions) est liée à la matrice $H = (X'X)^{-1}$ puisque $var(\hat{B}) = \sigma^2(X'X)^{-1}$. En pratique, lorsque les variables explicatives sont fortement corrélées entre elles ("problème de colinéarité"), ou lorsque le nombre de variables explicatives p est du même ordre de grandeur que n, alors certains coefficients de la matrice $(X'X)^{-1}$ deviennent grands (matrice 'mal conditionnée'), ce qui conduit à des estimateurs des moindres carrés avec une forte variance, et donc qui peuvent prendre des grandes valeurs.

Pour pouvoir estimer les paramètres lorsque la matrice $X'X$ n'est pas inversible ou réduire la variance de l'estimateur lorsque la matrice $X'X$ est mal conditionnée, la régression Ridge introduit un terme de pénalité qui permet de "pénaliser" les fortes valeurs de $\hat{B}$.

Définition 3 *On définit l'estimateur Ridge par*

$$\hat{B}_\lambda^{ridge} = \underset{B\in\mathbb{R}^{p+1}}{argmin}\{\|Y - XB\|^2 + \lambda \|B\|^2\}$$

avec $\lambda > 0$ un hyperparamètre à choisir (λ est appelé pénalité dans la suite).

Par définition, la régression Ridge va chercher un compromis pour obtenir un modèle qui décrit bien les données (le terme $\|Y - XB\|^2$ doit être "petit") avec des valeurs de paramètres qui ne doivent pas être trop grandes (le terme de régularisation $\lambda \|B\|^2$ doit aussi être "petit"). La valeur de la pénalité λ contrôle l'importance donnée au terme de régularisation. On vérifie aisément que

- si $\lambda = 0$ alors on retrouve l'estimateur des moindres carrés,
- si $\lambda \to +\infty$, alors $\hat{B}_\lambda^{ridge} \to 0$.

La proposition ci-dessous donne une expression explicite pour l'estimateur Ridge.

Proposition 3 *L'estimateur Ridge est donné par*

$$\hat{B}_\lambda^{ridge} = (X'X + \lambda I)^{-1}X'Y.$$

Preuve 2 *Posons* $F(B) = \|Y - XB\|^2 + \lambda \|B\|^2$. *On a*

$$F(B) = \|Y\|^2 - 2 < Y|XB > + \|XB\|^2 + \lambda \|B\|^2$$

et donc grad $F(B) = -2Y'X +' 2BX''X + 2\lambda B'$. *On déduit alors le résultat de l'expression grad* $F(\hat{B}_\lambda^{ridge}) = 0$.

La régression Ridge conduit donc à ajouter la matrice diagonale λI à la matrice $X'X$ avant de l'inverser, ce qui conduit généralement à inverser une matrice mieux conditionnée. Cette expression permet également de calculer le biais et la variance de l'estimateur Ridge et de vérifier que

- la variance de l'estimateur diminue lorsque λ augmente,
- la biais de l'estimateur augmente lorsque λ augmente.

Faire varier la valeur de l'hyperparamètre λ permet donc de chercher un compromis entre biais et variance. Le package *glmnet* qui sera utilisé dans les TD cherche le compromis optimal en faisant varier la valeur de λ sur une grille $\{\lambda_1, ..., \lambda_N\}$ puis en sélectionnant la valeur de λ qui donne la plus petite erreur de prédiction en validation croisée. Le choix de la grille $\{\lambda_1, ..., \lambda_N\}$ peut être délicat en pratique, cf exemples ci-dessous.

3.3.2 Régression Lasso

La régression Ridge est utile pour réduire la variance des estimateurs et améliorer les prédictions lorsque la matrice $X'X$ est mal conditionnée ou non-inversible. Un défaut de cette méthode est le manque d'interprétabilité puisque toutes les variables sont conservées dans le modèle (dit autrement, la régression Ridge ne permet pas de faire de la sélection de variables). La régression LASSO (Least Abslue Shrinkage ans Selection Operator) propose de résoudre ce problème en remplaçant la norme L^2 par la norme L^1 dans le terme de pénalité.

Définition 4 *On définit l'estimateur LASSO par*

$$\hat{B}_\lambda^{lasso} = \underset{B \in \mathbb{R}^{p+1}}{argmin} \{\|Y - XB\|^2 + \lambda \|B\|_1\}$$

avec $\lambda > 0$ *et* $\|B\|_1 = \sum_i |B_i|$.

Contrairement à l'estimateur Ridge, il n'existe pas d'expression explicite pour l'estimateur LASSO. En pratique, il faut donc utiliser un algorithme d'optimisation numérique pour résoudre ce problème d'optimisation. La fonction objectif n'étant pas différentiable (à cause du choix de la norme 1), les méthodes d'optimisation basées sur le gradient ne peuvent pas être utilisées. Le package *glmnet* utilise un algorithme de descente "par coordonnées" ("coordinate descent"). On résout alors successivement pour $i \in \{1, ..., p\}$ les problèmes d'optimisation univariés

$$B_i \mapsto F(B)$$

avec $F(B) = \|Y - XB\|^2 + \lambda \|B\|_1$. Comme pour la régression Ridge, le choix de la pénalité λ est fait par validation croisée.

Les estimateurs Ridge et Lasso sont solutions d'un problème dual ("version lagrangienne") donné dans la proposition suivante.

Proposition 4 *L'estimateur Ridge est solution du problème d'optimisation sous contrainte*

$$\hat{B}_\lambda^{ridge} = \underset{B \in \mathbb{R}^{p+1} | \|B\| < \tau^r}{argmin} \{\|Y - XB\|^2\}$$

avec $\tau^r > 0$ *qui dépend de* λ.
L'estimateur LASSO est solution du problème d'optimisation sous contrainte

$$\hat{B}_\lambda^{ridge} = \underset{B \in \mathbb{R}^{p+1} | \|B\|_1 < \tau^l}{argmin} \{\|Y - XB\|^2\}$$

avec $\tau^l > 0$ *qui dépend de* λ.

Preuve 3 *Posons* $\tau^r = \left\|\hat{B}_\lambda^{ridge}\right\|$. *Soit* $B \in \mathbb{R}^{p+1}$. *Par définition de l'estimateur Ridge, on a*

$$\|Y - XB\|^2 + \lambda \|B\|^2 \geq \left\|Y - X\hat{B}_\lambda^{ridge}\right\|^2 + \lambda \left\|\hat{B}_\lambda^{ridge}\right\|^2$$

Supposons de plus que $\|B\| < \tau^r$. *On a alors*

$$\|Y - XB\|^2 \geq \left\|Y - X\hat{B}_\lambda^{ridge}\right\|^2 + \lambda \left(\left\|\hat{B}_\lambda^{ridge}\right\|^2 - \|B\|^2\right) \geq \left\|Y - X\hat{B}_\lambda^{ridge}\right\|^2.$$

La preuve pour l'estimateur LASSO est similaire.

Cette proposition permet de comprendre pourquoi la régression LASSO conduit à sélectionner certaines variables au contraire de la régression Ridge (cf Figure 2).

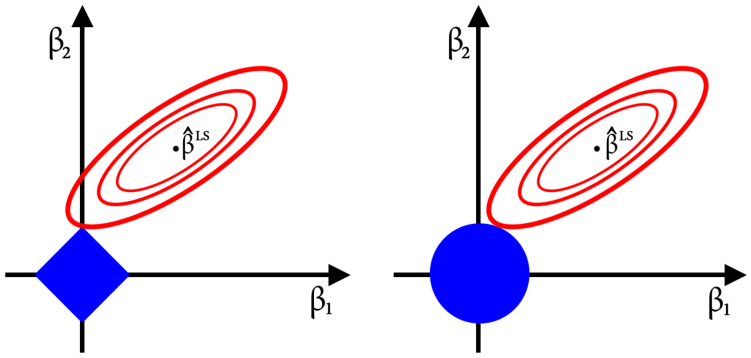


Figure 2: Comparaison des méthodes Ridge et LASSO. En bleu la zone où la contrainte est satisfaite, en rouge les lignes de niveau de la fonction $\beta \mapsto \|Y - X\beta\|^2$. $\hat{\beta}^{LS}$ représente l'estimateur des moindres carrés ("Least Square"). Source : https://fr.wikipedia.org/wiki/Lasso_(statistiques).

3.3.3 Elastic Net

Pour les TD, nous utiliserons le package *glmnet*. Une introduction à l'utilisation du package est disponible ici : https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html

Ce package résout le problème suivant (appelé Elastic Net dans la littérature)

$$\underset{B}{argmin}\{\|Y - XB\|^2 + \lambda \left((1-\alpha)\frac{\|B\|_2^2}{2} + \alpha \|B\|_1\right)\}$$

avec $\alpha \in [0,1]$ un paramètre qui contrôle le compromis entre la régression ridge ($\alpha = 0$) et la régression LASSO ($\alpha = 1$). L'intercept β_0 du modèle n'est pas inclus dans le terme de régularisation.

Régression ridge avec glmnet

```
###############Simulation des données#############
n=50;p=10  #50 individus et 10 variables explicatives
X=cbind(rep(1,n),matrix(rnorm(n*p),ncol=p))  #covariables
beta=c(rep(1,3),rep(0,p-2))  #coefficients : seuls les deux premières variables ont un effet
y=X%*%beta+rnorm(n)  #simulation de la réponse
###############Régression ridge#############
library(glmnet)
```

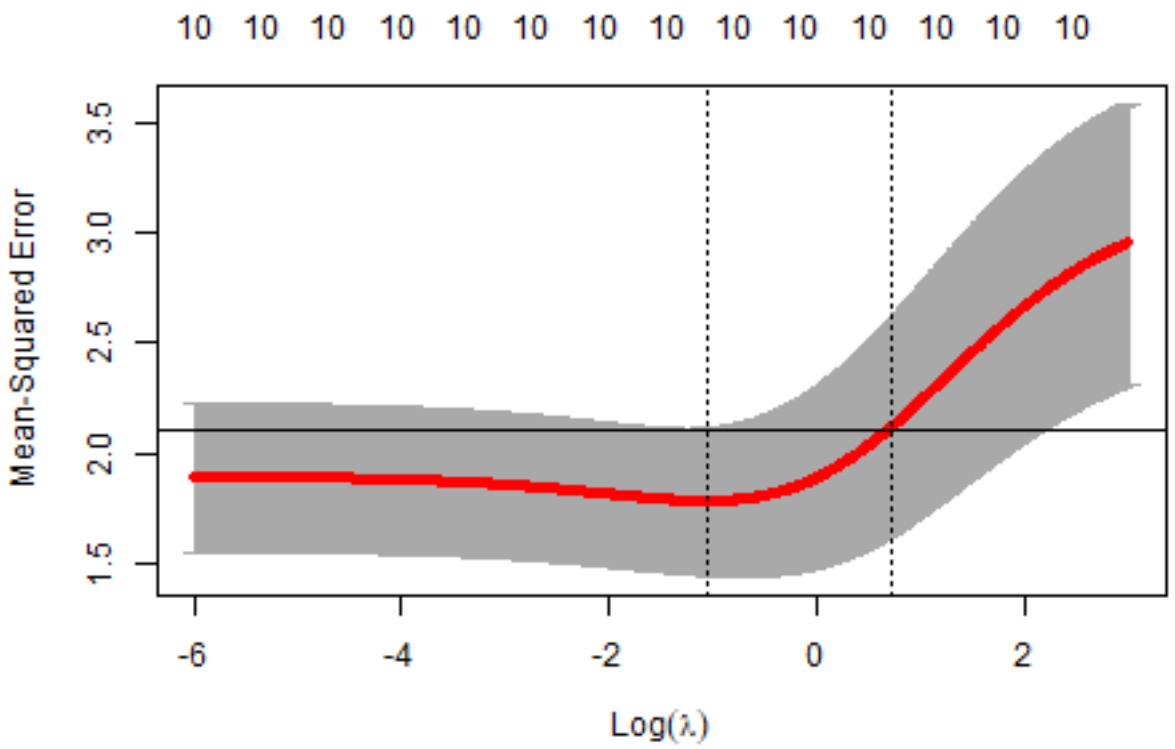
```
## Le chargement a nécessité le package : Matrix
```

```
## Loaded glmnet 4.1-4
```

```
#glmnet prend en entrée des matrices, pas un data.frame comme lm ou regsubsets
#Il ne faut pas que l'intercept soit dans X
X=X[,-1]

###############Validation croisée#############
#optimisation de lambda par validation croisée
tabl=exp(seq(-6,3,by=.01))  #Domaine pour la recherche d'un lambda optimal
#Rq : la grille pour lambda n'est pas toujours bien choisie par glmnet
#zoom sur les faibles valeurs de \lambda(pour retrouver EMC classique) + échelle log
cvfit = cv.glmnet(X, y,alpha=0,lambda=tabl,nfolds=length(y))
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
## fold
```

```
#nfolds=length(y) : "leave-one" cross-validation (nfolds est le nombre de blocs)
plot(cvfit) #RMSE en fonction de lambda
#La zone grisée correspond à l'intervalle +-std(MSE)
#avec std(MSE) l'écart-type de l'erreur quadratique moyenne
abline(h=cvfit$cvm[cvfit$lambda==cvfit$lambda.1se])
```




```
#lambda.1se est la valeur de lambda telle que la MSE égale est égale à
#la MSE minimum plus écart-type de l'erreur au point minimum
#Interprétation : régularisation plus forte mais pas de différence 'significative' en terme de MSE

#On peut extraire différentes quantités de cvfit
cvfit$lambda.min  #valeur optimale de lambda
```

```
## [1] 0.3464558
```

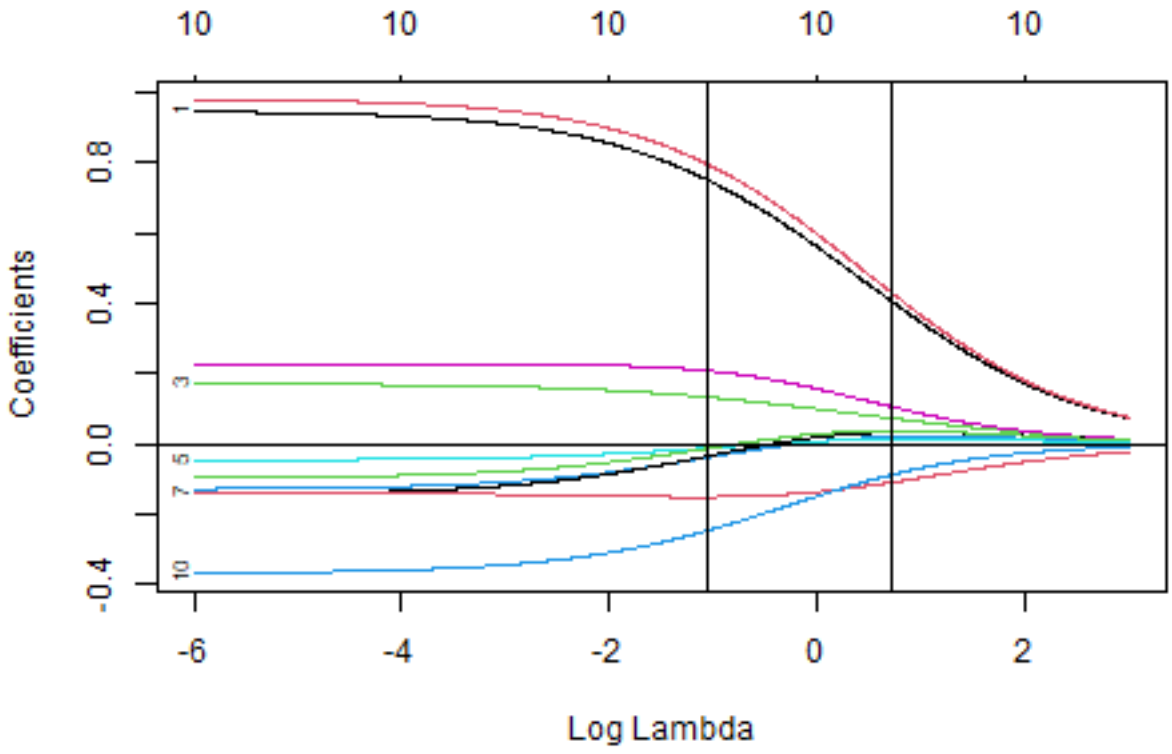
```r
coef(cvfit, s = "lambda.min")  #paramètres optimaux
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept)  0.90398482
## V1           0.74958135
## V2           0.79411241
## V3           0.13239316
## V4          -0.03884674
## V5          -0.01206983
## V6           0.20772670
## V7          -0.03477123
## V8          -0.15110922
## V9          -0.01307921
## V10         -0.24644451
```

```r
sqrt(cvfit$cvm[cvfit$lambda==cvfit$lambda.min])  #RMSE avec paramètres optimaux
```

```
##     s406
## 1.333875
```

```r
#Représentation des coefficients et des valeurs optimales
fit = glmnet(X, y,alpha=0,lambda=tabl)
plot(fit,xvar="lambda")
abline(v=log(cvfit$lambda.min))  #lambda avec RMSE minimum
abline(v=log(cvfit$lambda.1se))  #lambda avec RMSE égale à RMSE minimum plus écart-type de l'erreur au poin
abline(h=0)
#ajout de la légende sur l'axe de y
vnat=coef(fit)
vnat=vnat[-1,ncol(vnat)]
axis(2, at=vnat,line=-2,label=as.character(1:10),tick=FALSE, cex.axis=.7)  #Affichage nom des variables
```



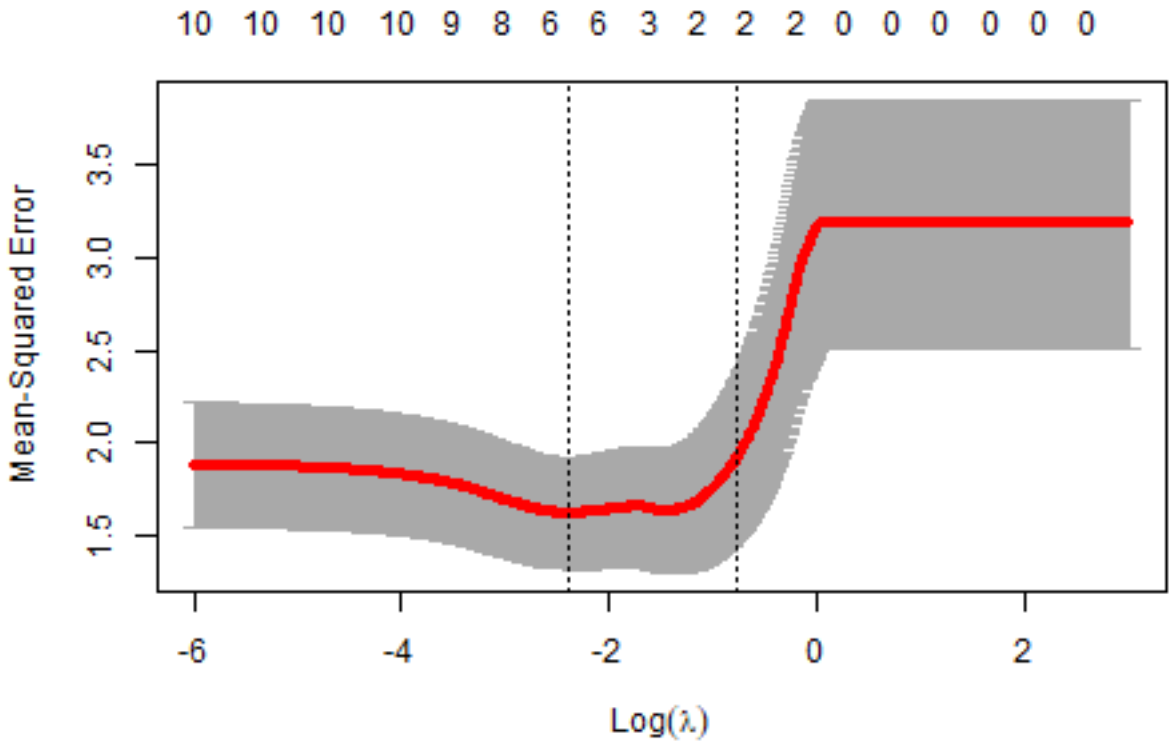

```
#Interprétation
#Lorsque lambda est petit, on retrouve EMC
#Tous les coefficients tendent vers 0 lorsque lambda tend vers +inf
```

Régression LASSO avec glmnet

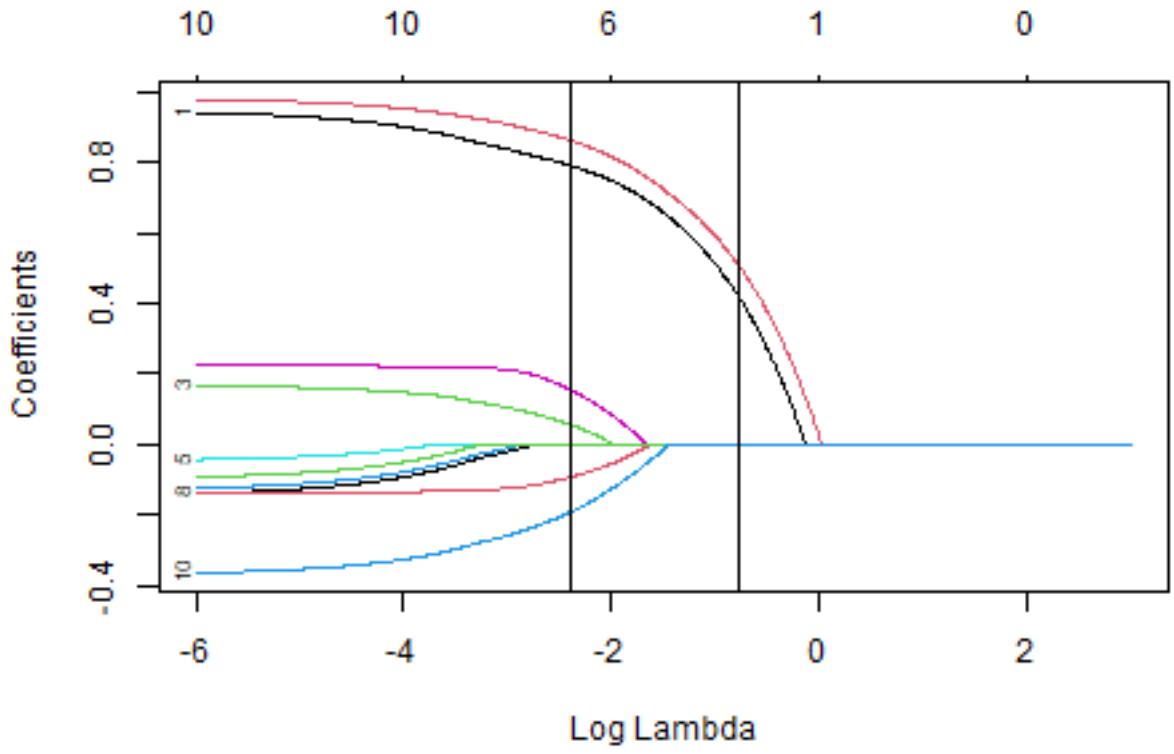
```
fit = glmnet(X, y,alpha=1,lambda=tabl)  #Régression LASSO : alpha=1
#optimisation de lambda par validation croisée
cvfit = cv.glmnet(X, y,alpha=1,lambda=tabl,nfolds=length(y))
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
## fold
```

```
#représentation de la MSE en fonction de lambda
plot(cvfit)
```




```
#représentation des coefficients en fonction de lambda
plot(fit,xvar="lambda")
abline(v=log(cvfit$lambda.min))
abline(v=log(cvfit$lambda.1se))
#ajout de la légende
vnat=coef(fit)
vnat=vnat[-1,ncol(vnat)]
axis(2, at=vnat,line=-2,label=as.character(1:10),tick=FALSE, cex.axis=.7,col='red') #Affichage nom des vari
```




```
#Interprétation
#Lorsque lambda est petit, on retrouve EMC
#Certains coefficients sont successivement mis à 0 lorsque lambda augmente
#Les coefficients associés aux variables 1 et 2 sont mis à 0 en dernier
#Au contraire de ridge, LASSO permet de faire de la sélection de variables
coef(cvfit, s = "lambda.min")  #paramètres optimaux
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept)  0.91242448
## V1           0.78927271
## V2           0.85998499
## V3           0.05580398
## V4           .
## V5           .
## V6           0.15387987
## V7           .
## V8          -0.09073416
## V9           .
## V10         -0.18970800
```

```
coef(cvfit, s = "lambda.1se") #paramètres régularisation plus forte
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                     s1
## (Intercept) 0.9007823
## V1          0.4136588
## V2          0.5037260
```

```
## V3          .
## V4          .
## V5          .
## V6          .
## V7          .
## V8          .
## V9          .
## V10         .
```

```
sqrt(cvfit$cvm[cvfit$lambda==cvfit$lambda.min])  #RMSE avec paramètres optimaux
```

```
##     s539
## 1.273687
```

```
#RMSE généralement plus faible avec LASSO que Ridge sur cet exemple
```

3.4 Régression sur variables qualitatives

Pour de nombreuses applications, certaines variables explicatives sont qualitatives ou discrètes et on doit alors adapter le modèle de régression linéaire. Le principe général consiste à recoder les variables qualitatives via des indicatrices en faisant attention aux problèmes de colinéarités.

3.4.1 Une seule variable qualitative : analyse de la variance à un facteur

L'analyse de la variance (ANOVA) à un facteur permet d'étudier l'effet d'une variable qualitative sur une variable quantitative.

Pour simplifier les notations, on note x_i la valeur prise par l'individu i et on recode les modalités de la variable explicative de telle manière que $x_i \in \{1, ..., p\}$ avec p le nombre de modalités de la variable qualitative. La variable explicative segmente alors les individus en p groupes et on notera n_j le nombre d'individus dans le groupe $j \in \{1, ..., p\}$.

Le **modèle d'analyse de la variance à un facteur** s'écrit

$$Y_i = \mu + \sum_{j=1}^{p} \alpha_j \mathbb{1}_{\{j\}}(x_i) + W_i.$$

avec $(W_1, ..., W_n)$ des variables aléatoires i.i.d. telles que $E[W_i] = 0$ et $var(W_i) = \sigma^2$ et $(\mu, \alpha_1, ..., \alpha_p, \sigma^2)$ des paramètres inconnus.

On retrouve un modèle de régression linéaire multiple dans lequel la variable qualitative initiale, avec p modalités, est codée par p variables explicatives quantitatives qui s'expriment comme des indicatrices à valeurs dans $\{0, 1\}$ ('dummy variables') avec seulement une variable qui prend une valeur différente de 0 pour un individu donné.

On vérifie aisément que

$$E[Y_i] = \mu + \sum_{j=1}^{p} \alpha_j \mathbb{1}_{\{j\}}(x_i)$$

et donc α_j s'interprète comme la différence (on parle d'"effet différentiel") entre l'espérance du groupe j et l'effet commun μ qui correspond à l'intercept du modèle de régression.

Sous forme matricielle, on obtient

$$Y = X\beta + W$$

avec $\beta = (\mu, \alpha_1, ..., \alpha_p)$,

$$X = \begin{pmatrix} 1 & \mathbb{1}_{\{1\}}(x_1) & \mathbb{1}_{\{2\}}(x_1) & \dots & \mathbb{1}_{\{p\}}(x_1) \\ 1 & \mathbb{1}_{\{1\}}(x_2) & \mathbb{1}_{\{2\}}(x_2) & \dots & \mathbb{1}_{\{p\}}(x_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \mathbb{1}_{\{1\}}(x_n) & \mathbb{1}_{\{2\}}(x_n) & \dots & \mathbb{1}_{\{p\}}(x_n) \end{pmatrix}$$

La matrice correspondant aux p dernières colonnes de X est appelée tableau disjonctif associé à la variable x.

Le modèle de l'ANOVA à un facteur est donc un modèle de régression linéaire multiple particulier et on peut utiliser les méthodes étudiées dans les chapitres précédents pour réaliser l'inférence statistique. Cependant, la matrice X n'est pas de rang $p+1$ puisque la première colonne est la somme des p dernières colonnes et la matrice $X'X$ n'est donc pas inversible (ce qui pose problème lorsqu'on utilise la méthode des moindres-carrés). Plus précisément, on peut vérifier que si toutes les modalités sont représentées (c'est à dire $n_j \geq 1$ pour $j \in \{1, ..., p\}$), alors la matrice X est de rang p. Afin de résoudre le problème de colinéarité (ou "d'identifiabilité" aussi appelé "dummy variable trap"), on peut enlever une colonne à la matrice X, ce qui revient à supposer qu'un des paramètres du modèle est égal à 0.

- Une première possibilité est de supposer que $\mu = 0$, c'est à dire à enlever la première colonne de la matrice X. On a alors
$$E[Y_i] = \sum_{j=1}^{p} \alpha_j \mathbb{1}_{\{j\}}(x_i)$$
et donc α_j s'interprète directement comme l'espérance du groupe j. Cette approche est utilisée par défaut par certains packages de machine learning en Python et R. Cependant, cela conduit à ajuster un modèle de régression sans intercept, ce qui conduit à des difficultés comme par exemple la définition et l'interprétation du coefficient R^2 ou du test de Fisher.

- Par défaut, R impose la **contrainte de type "cellule de référence"** $\alpha_1 = 0$, ce qui revient à supprimer la deuxième colonne de la matrice X. On a alors
$$E[Y_i] = \mu + \sum_{j=2}^{p} \alpha_j \mathbb{1}_{\{j\}}(x_i)$$
et donc $E[Y_i] = \mu$ si $x_i = 1$ et $E[Y_i] = \mu + \alpha_j$ si $x_i = j \geq 2$. On interprète donc μ comme l'espérance dans le groupe de référence (qui correspond à $x_i = 1$) et α_j comme un effet différentiel entre le groupe j et le groupe de référence.

Exemple. On considère un jeu de données avec $n = 4$ individus avec une variable explicative 'Genre' qui prend les valeurs suivantes $\{F, F, H, F\}$. Par défaut, R recode les variables en utilisant l'ordre alphabétique, ce qui donne la matrice
$$X = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$
. Avec la première contrainte, la matrice X devient
$$\tilde{X}_1 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$
et avec la deuxième contrainte, on obtient
$$\tilde{X}_2 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}.$$
On vérifie que les matrices $\tilde{X}_1$ et $\tilde{X}_2$ sont de rang 2 (et donc les matrices $\tilde{X}_1'\tilde{X}_1$ et $\tilde{X}_2'\tilde{X}_2$ sont inversibles) et que $Im(\tilde{X}_1) = Im(\tilde{X}_2)$ (et donc les deux contraintes conduisent à des modèles équivalents).

On supposera dans la suite du cours que la contrainte de type "cellule de référence" est utilisée. On peut alors expliciter l'estimateur des moindres carrés des paramètres (cf exercice 4 du TD). On notant $\bar{Y}_j = \frac{1}{n_j}\sum_{i=1}^{n} x_i \mathbb{1}_{\{j\}}(x_i) = \frac{1}{n_j}\sum_{i|x_i=j} x_i$ la moyenne empirique dans le groupe $j \in \{1, ..., p\}$, on peut montrer que

- l'estimateur des moindres carrés de μ est la moyenne empirique $\bar{Y}_1$ dans le groupe de référence,

- l'estimateur des moindres carrés de α_j est la différence $\bar{Y}_j - \bar{Y}_1$ entre la moyenne empirique dans le groupe j et la moyenne empirique dans le groupe de référence.

Afin de réaliser des tests ou faire des intervalles de confiance, on se place généralement sous les hypothèses du modèle linéaire gaussien et on suppose que $(W_1, ..., W_n) \sim_{iid} \mathcal{N}(0, \sigma^2)$. Ceci implique que les variables aléatoires $(Y_1, ..., Y_n)$ sont des variables aléatoires gaussiennes indépendantes de même variance mais avec une espérance qui dépend du groupe.

On peut alors en particulier tester l'hypothèse

$$H_0 : \alpha_2 = ... = \alpha_p = 0 \text{ contre } H_1 \ \exists i \neq j \text{ tel que } \mu_i \neq \mu_j$$

en utilisant le test d'analyse de la variance du paragraphe 3.1 (cas particulier 2).

L'hypothèse H_0 signifie que l'espérance de la variable à expliquer est la même dans toutes les groupes (test de comparaison de plusieurs espérances) et que la variable explicative n'a pas d'effet (en moyenne) sur la variable à expliquer. En utilisant les mêmes notations que dans le paragraphe~3.1, on peut montrer que le projeté orthogonal $\hat{Y}$ de Y sur $E = Im(X)$ est tel que $\hat{Y}_i = \sum_{i=1}^n \bar{Y}_j \mathbb{1}_{\{j\}}(x_i)$ puis que la statistique de test s'écrit

$$F = \frac{(n-p)SC_{reg}}{(p-1)SC_{res}}$$

avec

- $SC_{reg} = \sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2 = \sum_{j=1}^p n_j (\bar{Y}_j - \bar{Y})^2$ la somme des carrés expliquée par le facteur ("inter-classes")
- $SC_{res} = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 = \sum_{j=1}^p SC_j$ avec $SC_j = \sum_{i|x_i=j} (Y_i - \bar{Y}_j)^2$ la somme des carrés résiduelles à l'intérieur du groupe j ("intra-classes").

Si H_0 est vérifiée, alors $F \sim \mathcal{F}_{p-1,n-p}$ et on accepte H_0 avec un risque de première espèce α ssi $F < f_{p-1,n-p,1-\alpha}$.

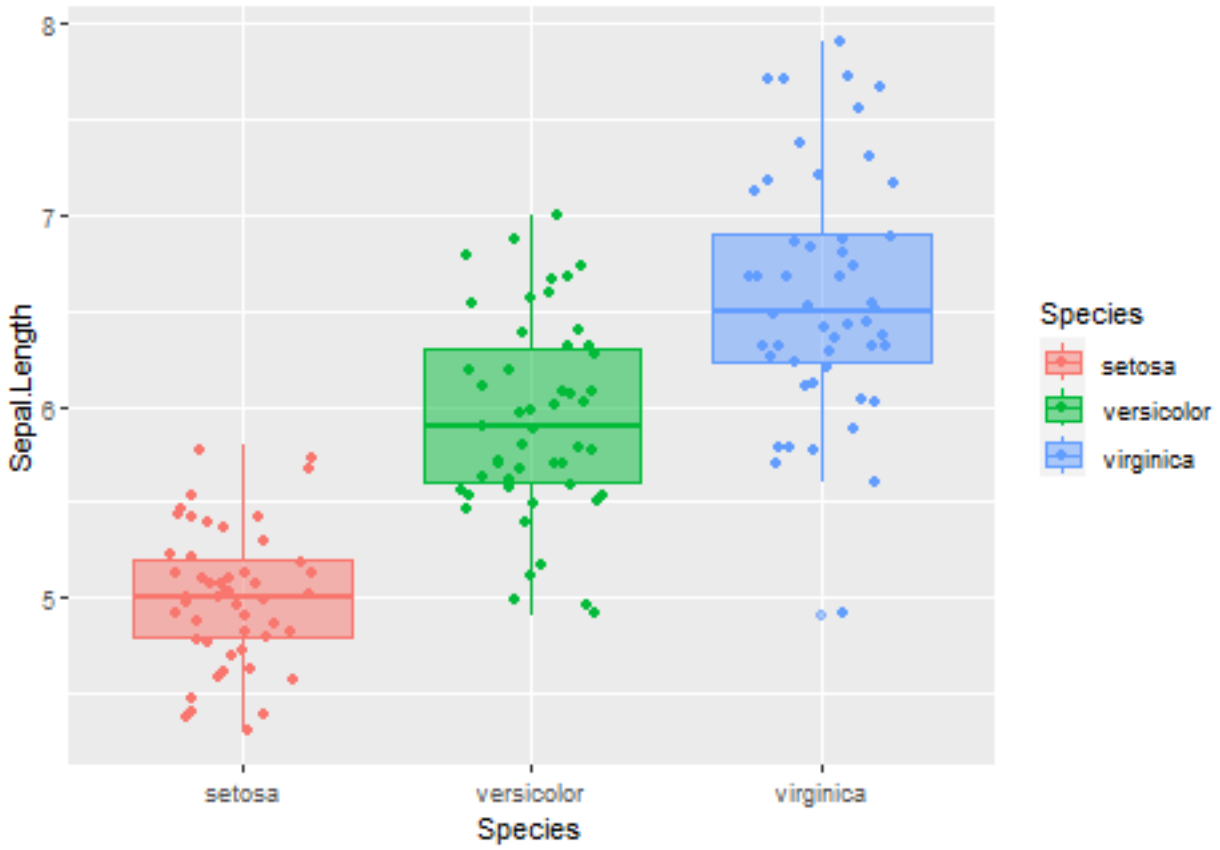
Interprétation : le théorème de Pythagore implique (formule "d'analyse de variance'') que

$$SC_{tot} = SC_{reg} + SC_{res}$$

avec $SC_{tot} = \sum_{i=1}^n (Y_i - \bar{Y})^2$ et la statistique de test de l'ANOVA est basée sur le rapport entre la variance expliquée par le facteur et la variance résiduelle : si ce rapport est petit alors le facteur apporte peu d'information et on accepte H_0.

Un exemple d'analyse de la variance à 1 facteur avec R. On considère le jeu de données classique *iris* disponible dans R.

```
library(ggplot2) #représentation graphique
p=ggplot(iris,aes(x=Species,y=Sepal.Length,colour=Species ,fill=Species))
p + geom_boxplot(alpha=.5)+geom_jitter(width=0.25)
```




```
summary(lm(Sepal.Length ~ Species, data = iris)) #ajustement et analyse du modèle
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Species, data = iris)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6880 -0.3285 -0.0060  0.3120  1.3120
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         5.0060     0.0728  68.762  < 2e-16 ***
## Speciesversicolor   0.9300     0.1030   9.033 8.77e-16 ***
## Speciesvirginica    1.5820     0.1030  15.366  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5148 on 147 degrees of freedom
## Multiple R-squared:  0.6187, Adjusted R-squared:  0.6135
## F-statistic: 119.3 on 2 and 147 DF,  p-value: < 2.2e-16
```

```
#Interprétation des paramètres
#Intercept décrit la moyenne dans le groupe de référence = setosa
#Speciesversicolor correspond à un effet différentiel entre groupe versicolor et groupe de référence
#la différence entre la moyenne des deux groupes est significative
#Speciesvirginica correspond à un effet différentiel entre groupe virginica et groupe de référence
#La p-value du test d'analyse de la variance est très faible
```

```r
#La différence entre la moyenne des différents groupes est significative
anova(lm(Sepal.Length ~ Species, data = iris)) #autre présentation du test
```

```
## Analysis of Variance Table
##
## Response: Sepal.Length
##            Df Sum Sq Mean Sq F value    Pr(>F)
## Species     2 63.212  31.606  119.26 < 2.2e-16 ***
## Residuals 147 38.956   0.265
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#La première ligne donne la SC inter-classe
#La deuxième ligne donne la SC intra-classe
```

3.4.2 Deux variables qualitatives : analyse de la variance à deux facteurs

L'analyse de la variance à 2 facteurs permet d'étudier l'impact de deux variables qualitatives $x_1 = (x_{1,1}, ..., x_{n,1})'$ et $x_2 = (x_{1,2}, ..., x_{n,2})'$ sur une variable quantitative $y = (y_1, ..., y_n)'$.

Une généralisation naturelle du modèle d'analyse de la variance à un facteur consisterait à recoder séparément les deux variables qualitatives en utilisant des indicatrices puis à ajuster un modèle de régression linéaire sur ces variables indicatrices (en faisant attention aux problèmes d'identifiabilité). Cependant, ceci ne permet pas de prendre en compte des éventuelles 'interactions' entre les deux variables explicatives. Par exemple, supposons que y_i représente les frais de santé de l'assuré i, $x_{1,i}$ sa catégorie d'âge et $x_{2,i}$ son métier. Il est possible que certains métiers conduisent à une augmentation plus importante des frais de santé en vieillissant que d'autres métiers : dans ce cas, il n'est pas raisonnable de supposer que les effets de l'âge et du métier s'ajoutent.

Notons p [resp. q] le nombre de modalités de la variable qualitative x_1 [resp x_2]. On recode ces variables de telle manière que $x_{k,1} \in \{1, ..., p\}$ et $x_{k,2} \in \{1, ..., q\}$. Les variables qualitatives segmentent les observations en $p * q$ groupes.

Le **modèle d'analyse de la variance à deux facteurs** s'écrit sous la forme d'un modèle de régression linéaire multiple

$$Y_i = \mu + \sum_{j=1}^{p} \alpha_j \mathbb{1}_{\{j\}}(x_{i,1}) + \sum_{k=1}^{q} \beta_k \mathbb{1}_{\{k\}}(x_{i,2}) + \sum_{j=1}^{p} \sum_{k=1}^{q} \gamma_{j,k} \mathbb{1}_{\{j\}}(x_{i,1}) \mathbb{1}_{\{k\}}(x_{i,2}) + W_i$$

avec un effet commun μ, des effets différentiels α_i selon la première variable, des effets différentiels β_j selon la deuxième variable et des termes d'interaction $\gamma_{i,j}$.

Le modèle s'écrit sous la forme matricielle suivante :

$$Y = \mu e + X_1 \alpha + X_2 \beta + X_{1,2} \gamma$$

avec $e = (1, ..., 1)'$, X_1 le tableau disjonctif associé à la variable x_1, $\alpha = (\alpha_1, ..., \alpha_p)'$, X_2 le tableau disjonctif associé à la variable x_2, $\beta = (\beta_1, ..., \beta_q)'$, $X_{1,2}$ la matrice de taille $(n, p \times q)$ de terme général $a_{k,(j-1)*p+i} = \mathbb{1}_{\{i\}}(x_{k,1}) \mathbb{1}_{\{j\}}(x_{k,2})$ pour $i \in \{1...p\}$ et $j \in \{1...q\}$, et $\gamma = (\gamma_{1,1}, \gamma_{2,1}, ..., \gamma_{p,1}, \gamma_{1,2}, ..., \gamma_{p,q})'$. On a également une écriture de la forme

$$Y = X\theta + W$$

avec $\theta = (\mu, \alpha', \beta', \gamma')'$ et $X = (e, X_1, X_2, X_{1,2}) \in \mathcal{M}_{n,(p+1)(q+1)}$.

Comme pour l'ANOVA à un facteur, on peut facilement vérifier que la matrice X n'est pas de rang plein (elle est de rang $p \times q$). Il faut alors ajouter $p + q + 1$ contraintes d'indentifiabilité sur les paramètres pour éviter les problèmes de colinéarité. R impose par défaut des contraintes de type "cellule de référence''

$$\begin{aligned}
&\alpha_1 = 0 \\
&\beta_1 = 0 \\
&\gamma_{i,1} = 0 \ \forall i \in \{1...p\} \\
&\gamma_{1,j} = 0 \ \forall j \in \{1...q\}.
\end{aligned}$$

Imposer ces contraintes revient à supprimer $p+q+1$ colonnes de la matrice X et rend la matrice des covariables de rang plein. On peut alors utiliser les outils vus pour le modèle linéaire. Les paramètres du modèle s'interprètent comme des effets différentiels par rapport à un individu de référence.

On suppose en outre dans la suite du paragraphe que $(W_1, ..., W_n) \sim_{iid} \mathcal{N}(0, \sigma^2)$. On peut alors utiliser le test d'analyse de la variance pour tester des modèles réduits. On commence généralement par tester la significativité de l'interaction, c'est à dire:

$$H_0 : \gamma_{j,k} = 0 \ \forall j \in \{1...p\}, k \in \{1...q\} \text{ contre } H_1 : \exists j, k \ \gamma_{j,k} \neq 0.$$

Sous H_0, le modèle réduit s'écrit donc

$$Y_i = \mu + \sum_{i=1}^{p} \alpha_j \mathbb{1}_{\{j\}}(x_{i,1}) + \sum_{k=1}^{q} \beta_k \mathbb{1}_{\{k\}}(x_{i,2}) + W_i.$$

On suppose alors que les effets des 2 variables sont additifs, et qu'il n'y a pas d'effet supplémentaire induit par l'interaction entre les deux variables.

Si H_0 est refusée, alors les deux variables ont un effet, et on peut pas supposer que les effets s'ajoutent. Si H_0 est acceptée, on teste ensuite si chacune des deux variables a un effet. Par exemple, pour la première variable, on teste:

$$H_0 : \alpha_1 = ... = \alpha_p = 0 \text{ contre } H_1 \ \exists i \text{ tel que } \alpha_i \neq 0$$

Sous H_0, le modèle réduit s'écrit donc

$$Y_i = \mu + \sum_{k=1}^{q} \beta_k \mathbb{1}_{\{k\}}(x_{i,2}) + W_i.$$

On peut également tester la significativité de l'effet simultané des deux variables, c'est à dire l'hypothèse:

$$H_0 : \alpha_1 = ... = \alpha_p = \beta_1 = ... = \beta_q = 0 \text{ contre } H_1 \ \exists i \text{ tel que } \alpha_i \neq 0 \text{ ou } \beta_i \neq 0.$$

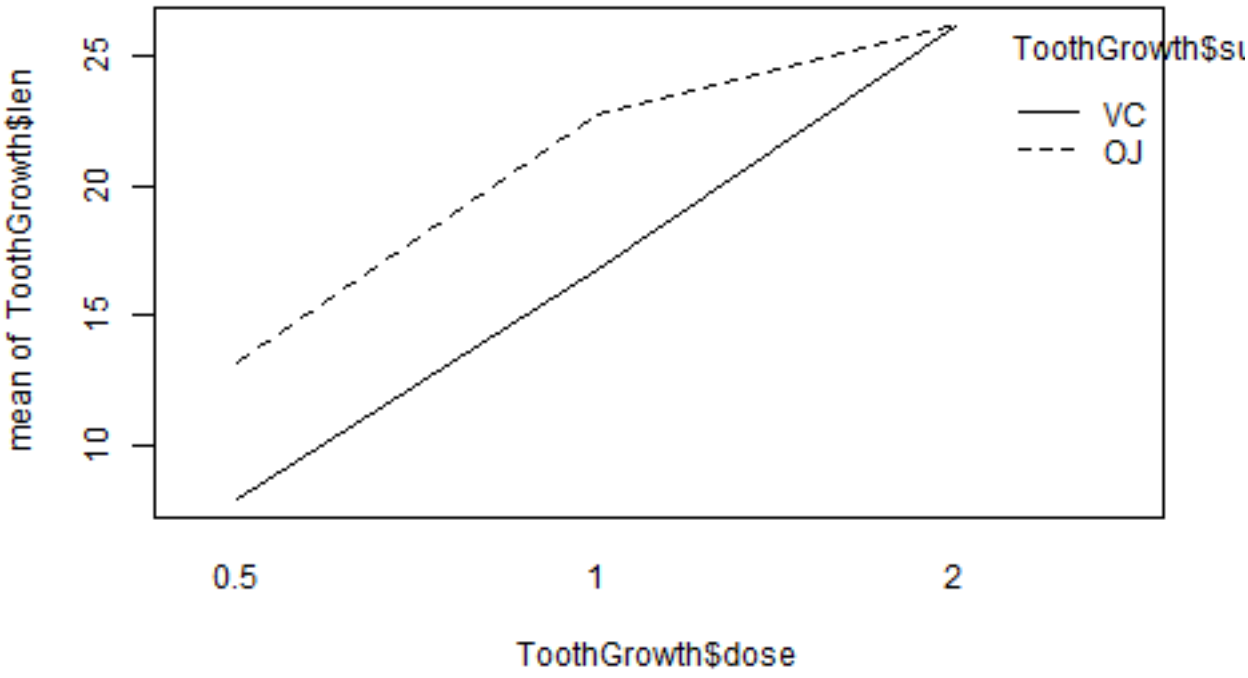
Généralisation. L'analyse de la variance à 2 facteurs se généralise à plus de 2 facteurs. Le principe est le même : on recode les variables explicatives par des indicatrices, on rajoute éventuellement les interactions à 2, 3, ... facteurs et les contraintes d'identifiabilité adaptées.

Un exemple d'analyse de la variance à 2 facteurs avec R. On considère le jeu de données *ToothGrowth* disponible dans R. Il donne la longueur d'une cellule responsable de la croissance des dents (variable quantitative *len*) pour 60 Cochons d'Inde en fonction de la dose de vitamine C (variable qualitative *dose* à valeurs dans 0.5, 1 et 2) et de la manière dont elle est administrée (variable qualitative *supp* à valeurs dans VC ou OJ).

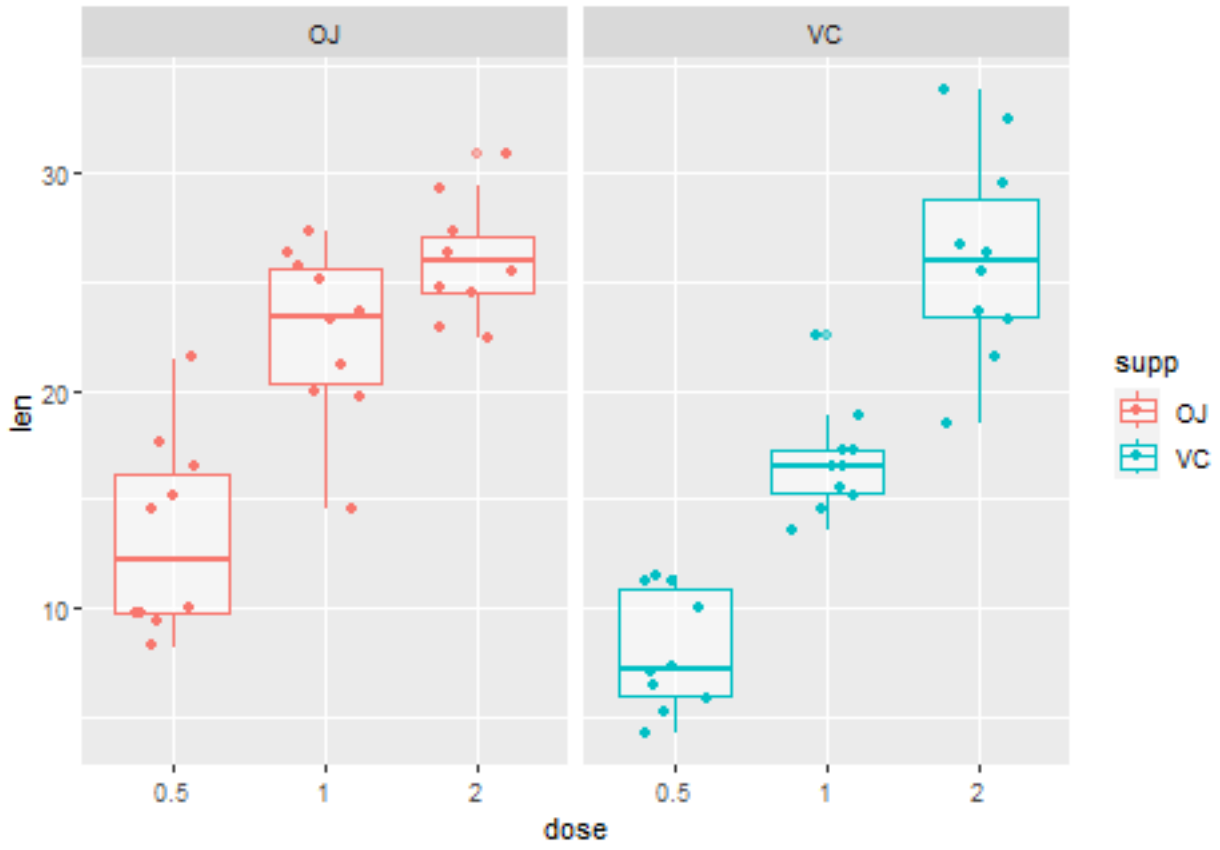
```
head(ToothGrowth)
```

```
##    len supp dose
## 1  4.2   VC  0.5
## 2 11.5   VC  0.5
## 3  7.3   VC  0.5
## 4  5.8   VC  0.5
## 5  6.4   VC  0.5
## 6 10.0   VC  0.5
```

```
ToothGrowth$dose=as.factor(ToothGrowth$dose)  #transformation de la variable dose en variable qualitative
interaction.plot(ToothGrowth$dose,ToothGrowth$supp,ToothGrowth$len) #représentation graphique
```




```
#Les lignes ne semblent pas parallèle : existence d'un interaction?
#Représentation alternative avec ggplot
ggplot(ToothGrowth,aes(x = dose, y = len,colour=supp)) +
  geom_boxplot(alpha=.5) +
  geom_jitter(width=0.25)+
  facet_wrap(~supp)
```




```
#L'augmentation de la dose de 1 à 2 ne semble pas avoir le même effet pour les deux modes d'administration
fit1=lm(len~dose*supp,data=ToothGrowth) #ajustement du modèle avec interaction
#L'interaction se code avec : ou avec *
#Taper ? formula dans R pour plus d'informations
summary(fit1)  #Il faut savoir interpréter les paramètres du modèle
```

```
##
## Call:
## lm(formula = len ~ dose * supp, data = ToothGrowth)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8.20  -2.72  -0.27   2.65   8.27
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    13.230      1.148  11.521 3.60e-16 ***
## dose1           9.470      1.624   5.831 3.18e-07 ***
## dose2          12.830      1.624   7.900 1.43e-10 ***
## suppVC         -5.250      1.624  -3.233  0.00209 **
## dose1:suppVC   -0.680      2.297  -0.296  0.76831
## dose2:suppVC    5.330      2.297   2.321  0.02411 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.631 on 54 degrees of freedom
## Multiple R-squared:  0.7937, Adjusted R-squared:  0.7746
## F-statistic: 41.56 on 5 and 54 DF,  p-value: < 2.2e-16
```

```r
#L'individu de référence a pour modalité dose=05 et supp=OJ
#Les coefficients s'interprètent comme des effets différentiels par rapport à la référence
#Par exemple, l'espérance pour un individu avec dose=2 et supp=VC est modélisée par
# la somme des paramètres Intercept+ dose2+suppVC+dose2:suppVC
fit0=lm(len~dose+supp,data=ToothGrowth) #ajustement du modèle sans interaction
summary(fit0)
```

```
##
## Call:
## lm(formula = len ~ dose + supp, data = ToothGrowth)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.085 -2.751 -0.800  2.446  9.650
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.4550     0.9883  12.603  < 2e-16 ***
## dose1         9.1300     1.2104   7.543 4.38e-10 ***
## dose2        15.4950     1.2104  12.802  < 2e-16 ***
## suppVC       -3.7000     0.9883  -3.744 0.000429 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.828 on 56 degrees of freedom
## Multiple R-squared:  0.7623, Adjusted R-squared:  0.7496
## F-statistic: 59.88 on 3 and 56 DF,  p-value: < 2.2e-16
```

```r
#Les termes qui modélisent l'interaction ont disparu
#Par exemple, l'espérance pour un individu avec dose=2 et supp=VC est modélisée par
# la somme des paramètres Intercept+ dose2+suppVC
anova(fit0,fit1)  #H0 refusée pour alpha=5% : l'interaction est significative
```

```
## Analysis of Variance Table
##
## Model 1: len ~ dose + supp
## Model 2: len ~ dose * supp
##   Res.Df    RSS Df Sum of Sq     F  Pr(>F)
## 1     56 820.43
## 2     54 712.11  2    108.32 4.107 0.02186 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.4.3 Mélange de variables quantitatives et qualitatives : analyse de la covariance

On suppose dans ce paragraphe qu'on cherche à analyser l'effet simultané de 2 variables explicatives x et z sur la réponse Y avec :

- $x = (x_1, ..., x_n)$ une variable quantitative continue.
- $z = (z_1, ..., z_n)$ une variable qualitative à p modalités. Comme dans les paragraphes précédents, on suppose que $z_k \in \{1, ..., p\}$.

Une démarche naturelle consiste alors à ajuster p modèles de régression, un pour chaque modalité de z. Le **modèle d'analyse de la covariance** s'écrit

$$Y_i = \alpha + \gamma x_i + \sum_{j=1}^{p} (\alpha_j + \gamma_j x_i) \mathbb{1}_{\{j\}}(z_i) + W_i.$$

Ce modèle introduit une ordonnée à l'origine α et une pente β commune et des effets différentiels pour chaque modalité de la variable qualitative.

Afin de rendre le modèle identifiable, R impose la contrainte

$$\alpha_1 = \gamma_1 = 0.$$

Avec cette contrainte, on a

$$E[Y_i] = \begin{cases} \alpha + \gamma x & \text{si } z_i = 1 \\ (\alpha + \alpha_j) + (\gamma + \gamma_j)x & \text{si } z_i = j > 1 \end{cases}.$$

Avec ces contraintes, le modèle s'écrit sous la forme d'un modèle linéaire $Y = X\beta + W$ avec X de rang plein. Les coefficients de la matrice X s'expriment en fonction de x_i, $\mathbb{1}_{\{j\}}(z_i)$ et $x_i\mathbb{1}_{\{j\}}(z_i)$. On peut alors utiliser les méthodes d'inférence statistique étudiées dans les chapitres précédents. En particulier, on peut utiliser l'analyse de la variance pour tester des modèles réduits. Ces tests sont valides si $(W_1, ..., W_n) \sim_{iid} \mathcal{N}(0, \sigma^2)$.

Classiquement, on teste (selon l'application considérée) :

1. L'égalité des pentes

   $$H_0 : \ \gamma_1 = ... = \gamma_p = \gamma \text{ contre } H_1 : \ \exists i, j \ \gamma_i \neq \gamma_j$$

   Sous H_0, le modèle s'écrit donc

   $$Y_i = \alpha + \gamma x_i + \sum_{j=1}^{p} \alpha_j \mathbb{1}_{\{j\}}(z_i) + W_i.$$

   Il n'y a alors plus d'"interaction" entre x et z.

2. L'égalité des ordonnées à l'origine

   $$H_0 : \ \alpha_1 = ... = \alpha_p = \alpha \text{ contre } H_1 : \ \exists i, j \ \alpha_i \neq \alpha_j$$

   Sous H_0, le modèle s'écrit donc

   $$Y_{i,j} = \alpha + \gamma_j x_{i,j} + W_{i,j}.$$

   $$Y_i = \alpha + \gamma x_i + \sum_{j=1}^{p} \gamma_j x_i \mathbb{1}_{\{j\}}(z_i) + W_i.$$

3. L'égalité des pentes et des ordonnées à l'origine:

   $$H_0 : \ \alpha_1 = ... = \alpha_p = \alpha \ et \ \gamma_1 = ... = \gamma_p = \gamma \text{ contre } H_1 : \ \exists i, j \ \alpha_i \neq \alpha_j \ ou \gamma_i \neq \gamma_j$$

   Sous H_0, le modèle s'écrit donc

   $$Y_i = \alpha + \gamma x_i + W_i.$$

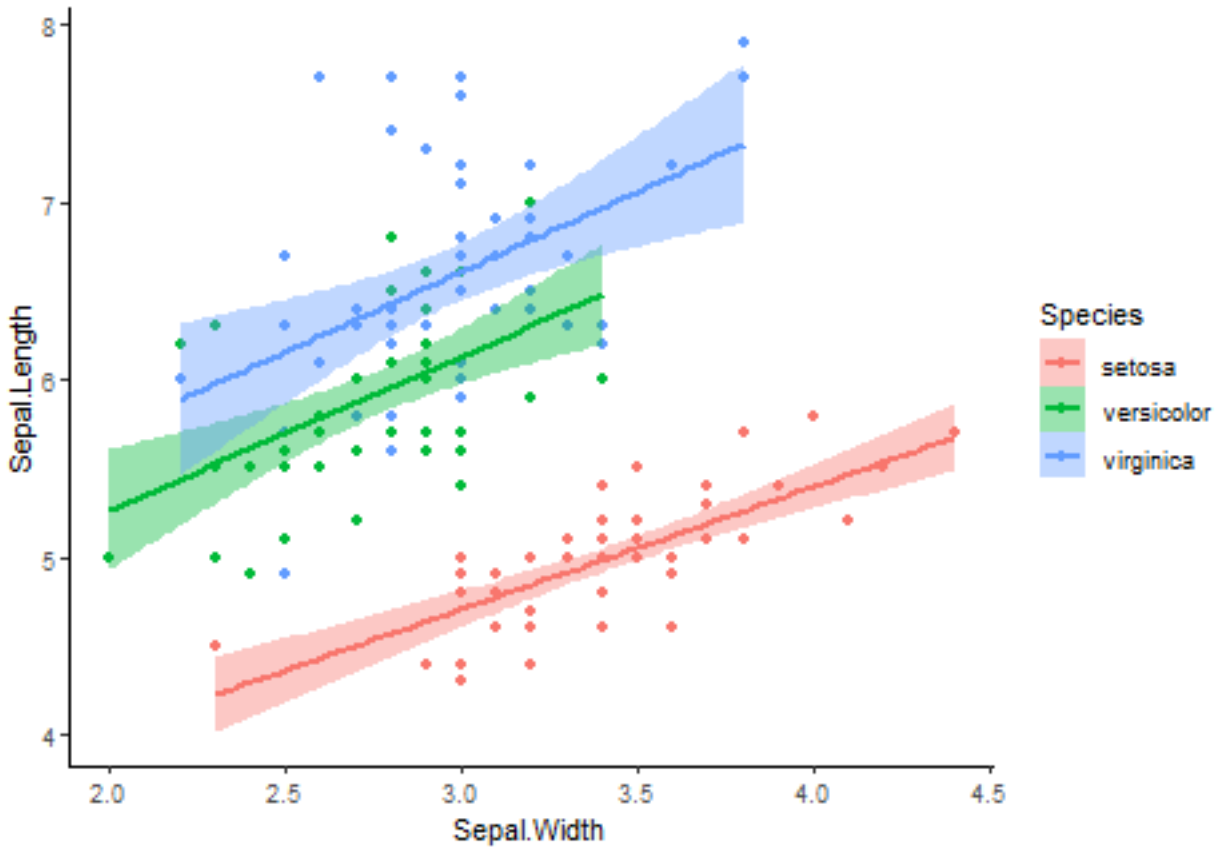
   Seule la variable x a alors un effet sur la réponse Y.

Généralisation. L'analyse de la covariance se généralise à plus de 2 variables. Le principe est le même : on recode les variables qualitatives par des indicatrices, on rajoute éventuellement les interactions et les contraintes d'identifiabilité adaptées.

Un exemple d'analyse de la covariance avec R.

```r
#Représentation graphique avec ggplot2
ggplot(iris, aes(y=Sepal.Length, x=Sepal.Width,colour=Species ,fill=Species))+
  geom_point()+
  geom_smooth(method = "lm")+
  theme_classic()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
#On veut expliquer la variable Sepal.Length à partir des variables Sepal.Width (quantitative) et  Species (
#On voit les droites de régression ajustées pour les trois espèces d'iris
#Les zones colorées représentent des intervalles de confiance

fit1=lm(Sepal.Length~Sepal.Width*Species,data=iris)
summary(fit1)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Sepal.Width * Species, data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.26067 -0.25861 -0.03305  0.18929  1.44917
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    2.6390     0.5715   4.618 8.53e-06 ***
## Sepal.Width                    0.6905     0.1657   4.166 5.31e-05 ***
## Speciesversicolor              0.9007     0.7988   1.128    0.261
```

```
## Speciesvirginica                1.2678     0.8162   1.553    0.123
## Sepal.Width:Speciesversicolor   0.1746     0.2599   0.672    0.503
## Sepal.Width:Speciesvirginica    0.2110     0.2558   0.825    0.411
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4397 on 144 degrees of freedom
## Multiple R-squared:  0.7274, Adjusted R-squared:  0.718
## F-statistic: 76.87 on 5 and 144 DF,  p-value: < 2.2e-16
```

```
#il faut savoir écrire mathématiquement le modèle ajusté
#et interpréter les coefficients
#Utilisons des tests pour vérifier si un modèle réduit est adapté
fit0=lm(Sepal.Length~Sepal.Width+Species,data=iris)
#modèle avec égalité des pentes
summary(fit0)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Sepal.Width + Species, data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.30711 -0.25713 -0.05325  0.19542  1.41253
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2.2514     0.3698   6.089 9.57e-09 ***
## Sepal.Width         0.8036     0.1063   7.557 4.19e-12 ***
## Speciesversicolor   1.4587     0.1121  13.012  < 2e-16 ***
## Speciesvirginica    1.9468     0.1000  19.465  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.438 on 146 degrees of freedom
## Multiple R-squared:  0.7259, Adjusted R-squared:  0.7203
## F-statistic: 128.9 on 3 and 146 DF,  p-value: < 2.2e-16
```

```
anova(fit0,fit1)
```

```
## Analysis of Variance Table
##
## Model 1: Sepal.Length ~ Sepal.Width + Species
## Model 2: Sepal.Length ~ Sepal.Width * Species
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    146 28.004
## 2    144 27.846  2   0.15719 0.4064 0.6668
```

```
#on accepte H0 : on peut supposer que les pentes sont égales
fit0=lm(Sepal.Length~Sepal.Width:Species,data=iris)
#modèle avec trois pentes différentes mais même intercept
summary(fit0)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Sepal.Width:Species, data = iris)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.16792 -0.26023 -0.03635  0.18797  1.52368
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    3.35789    0.33000  10.175  < 2e-16 ***
## Sepal.Width:Speciessetosa      0.48326    0.09683   4.991 1.69e-06 ***
## Sepal.Width:Speciesversicolor  0.92991    0.11976   7.765 1.32e-12 ***
## Sepal.Width:Speciesvirginica   1.08401    0.11166   9.708  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4406 on 146 degrees of freedom
## Multiple R-squared:  0.7226, Adjusted R-squared:  0.7169
## F-statistic: 126.8 on 3 and 146 DF,  p-value: < 2.2e-16
```

```
anova(fit0,fit1)
```

```
## Analysis of Variance Table
##
## Model 1: Sepal.Length ~ Sepal.Width:Species
## Model 2: Sepal.Length ~ Sepal.Width * Species
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    146 28.345
## 2    144 27.846  2   0.49817 1.2881  0.279
```

```
#on accepte H0 : on peut supposer que les intercept sont égaux
fit0=lm(Sepal.Length~Sepal.Width,data=iris)
#modèle avec pentes et intercept égaux
summary(fit0)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Sepal.Width, data = iris)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5561 -0.6333 -0.1120  0.5579  2.2226
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.5262     0.4789   13.63   <2e-16 ***
## Sepal.Width  -0.2234     0.1551   -1.44    0.152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8251 on 148 degrees of freedom
## Multiple R-squared:  0.01382,    Adjusted R-squared:  0.007159
## F-statistic: 2.074 on 1 and 148 DF,  p-value: 0.1519
```

```
anova(fit0,fit1)
```

```
## Analysis of Variance Table
##
```

```
## Model 1: Sepal.Length ~ Sepal.Width
## Model 2: Sepal.Length ~ Sepal.Width * Species
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1    148 100.756
## 2    144  27.846  4     72.91 94.258 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#on refuse H0
#Conclusion : on ne peut pas supposer que les intercept ET les pentes sont égaux simultanément
```

4 Introduction aux modèles linéaires généralisés courament utilisés en actuariat

4.1 Introduction

Dans le modèle linéaire gaussien, on suppose que les variables aléatoires $Y_1, ..., Y_n$ sont **indépendantes** et que

$$Y_i \sim \mathcal{N}(\beta_0 + \beta_1 x_{i,1} + ... + \beta_p x_{i,p}, \sigma^2)$$

$\mu(x_{i,1}, ..., x_{i,p}) = \beta_0 + \beta_1 x_{i,1} + ... + \beta_p x_{i,p}$. En particulier, dans le modèle linéaire gaussien la variable à prédire Y_i est supposée être à valeurs continues dans $\mathbb{R}$ (le support de la loi normale est $\mathbb{R}$). Ils ne sont donc pas adaptés pour modéliser des variables à expliquer qui sont qualitatives, discrètes ou à valeurs dans une sous partie de $\mathbb{R}$. Pourtant, ce type de variable est courant en actuariat. Par exemple, un actuaire peut être amené à modéliser une variable Y_i

- **binaire**, à valeurs dans $\{0, 1\}$, par exemple décrivant la présence/absence d'un sinistre, d'une maladie ou d'une fraude,
- **discrète**, à valeurs dans $\mathbb{N}$, telle que le nombre de sinistres,
- **positive**, à valeurs dans $\mathbb{R}^+$, par exemple le montant des sinistres ou une durée de vie.

Le modèle linéaire gaussien n'est plus adapté dans ce genre de situation. Les modèles linaires généralisés (GLM pour "Generalized Linear Model") fournissent alors une alternative.

On supposera dans la suite que les p variables explicatives sont quantitatives. Le cas des variables explicatives qualitatives se traite de la même manière que pour la régression linéaire ('recodage' via les indicatrices avec prise en compte des colinéarités).

4.2 GLM courament utilisés en actuariat

Dans le cadre de ce cours, on s'intéresse uniquement aux GLM couramment utilisés en actuariat. Il est possible de construire plus généralement les GLM pour la famille de loi exponentielle qui inclut comme cas particuliers la loi normale, la loi de Bernoulli, la loi de Poisson et la loi Gamma sur lesquelles nous allons nous focaliser dans la suite. On pourra consulter le livre de Dobson et al. (2018) pour plus de détails.

La structure d'un GLM repose sur les trois hypothèses décrites ci-dessous.

1. On suppose que les variables aléatoires $Y_1, ..., Y_n$ sont **indépendantes**.
2. On suppose la loi de Y_i appartient à **une famille de distribution paramétrique** dont les paramètres dépendent des variables explicatives. Les modèles les plus classiques en actuariat sont décrits ci-dessous.

(a) **Modèle GLM gaussien.** On suppose que

$$Y_i \sim \mathcal{N}\left(\mu(x_{i,1},...,x_{i,p}), \sigma^2\right).$$

On remarque qu'on retrouve le modèle linéaire gaussien dans le cas particulier où $\mu(x_{i,1},...,x_{i,p}) = \beta_0 + \beta_1 x_{i,1} + ... + \beta_p x_{i,p}$.

(b) **Modèle GLM Bernoulli.** On suppose que

$$Y_i \sim \mathcal{B}er\left(\pi(x_{i,1},...,x_{i,p})\right).$$

Ce modèle est utilisé pour modéliser des variables à expliquer binaires, le paramètre de la loi de Bernoulli $P(Y_i = 1) = \pi(x_{i,1},...,x_{i,p})$ dépend des variables explicatives. On a alors $E[Y_i] = \pi(x_{i,1},...,x_{i,p})$.

(c) **Modèle GLM Poisson.** On suppose que

$$Y_i \sim Pois(\lambda(x_{i,1},...,x_{i,p})).$$

Ce modèle est couramment utilisé en actuariat pour décrire des variables à expliquer à valeurs dans $\mathbb{N}$ (nombre de sinistres notamment). On rappelle qu'une variable aléatoire Y à valeurs dans $\mathbb{N}$ suit une loi de Poisson de paramètre λ si

$$P[Y = k] = exp(-\lambda)\frac{\lambda^k}{k!}$$

On a alors $E[Y] = var(Y) = \lambda$: l'espérance et la variance d'une loi de Poisson sont égales, ce qui peut être limitation de ces modèles (la dispersion est directement liée à la moyenne).

(d) **Modèle GLM gamma.** On suppose que

$$Y_i \sim Gam(\alpha, \sigma(x_{i,1},...,x_{i,p})).$$

Ce modèle est couramment utilisé en actuariat pour décrire des variables à valeurs positives (montant des sinistres notamment). On rappelle qu'une variable aléatoire Y à valeurs dans $\mathbb{R}^+$ suit une loi gamma de paramètres $(\alpha, \sigma) \in (\mathbb{R}^{+*})^2$ si sa densité est

$$f(x;\alpha,\sigma) = \frac{1}{\sigma^\alpha \Gamma(\alpha)} x^{\alpha-1} \exp(-\frac{x}{\sigma}) \mathbb{1}_{]0,+\infty[}(x).$$

α est un paramètre de forme et σ un paramètre d'échelle. Si $X \sim \gamma(\alpha,\sigma)$, alors $E[X] = \alpha\sigma$ et $var(X) = \alpha\sigma^2 = \frac{E[X]^2}{\alpha}$.

3. On modélise la relation entre les variables explicatives et la variable à expliquer en utilisant **une fonction lien** g. Plus précisément, g permet de lier l'espérance de la loi de Y_i à une combinaison linéaire des variables explicatives via l'équation

$$g(E[Y_i]) = \beta_0 + \beta_1 x_{i,1} + ... + \beta_p x_{i,p}$$

A chaque famille de loi est associé lien par défaut (ou lien "canonique"), mais il est possible d'adapter g selon le jeu de données considéré. La commande `R ? family` donne les liens canoniques utilisés par R. En particulier, pour les modèles discutés ci-dessus, on obtient

gaussian(link = "identity") : le lien par défaut associé à la loi de Bernoulli est le lien identité.

binomial(link = "logit") : le lien par défaut associé à la loi de Bernoulli est le lien logit.

poisson(link = "log") : : le lien par défaut associé à la loi de Poisson est la fonction ln.

Gamma(link = "inverse") : le lien par défaut associé à la loi Gamma est la fonction inverse.

Ces modèles sont discutés plus précisément ci-dessous.

(a) **Modèle GLM gaussien.** Le lien par défaut est la fonction identité ($g(x) = x$) et on retrouve alors le modèle linéaire gaussien.

(b) **Modèle GLM Bernoulli.** Le lien par défaut est la fonction

$$logit(u) = ln(\frac{u}{1-u}).$$

La fonction *logit* est une bijection $]0,1[\to\mathbb{R}$ et permet donc de transformer le paramètre $\pi\in]0,1[$ de la loi de Bernoulli en un nombre réel. Le modèle GLM Bernoulli avec lien *logit* est appelé **modèle de régression logistique**. Dans ce modèle, on a

$$logit(E[Y_i]) = ln\left(\frac{P[Y_i=1]}{1-P[Y_i=1]}\right) = ln\left(\frac{P[Y_i=1]}{P[Y_i=0]}\right) = \beta_0+\beta_1x_{i,1}+...+\beta_px_{i,p}.$$

On en déduit que

$$\frac{P[Y_i=1]}{P[Y_i=0]} = exp(\beta_0+\beta_1x_{i,1}+...+\beta_px_{i,p})$$

avec $\frac{P[Y_i=1]}{P[Y_i=0]}$ le rapport entre la probabilité de réussite et la probabilité d'échec ("odds ratio"). On peut interpréter $exp(\beta_j)$ comme le taux d'augmentation du odds ratio lorsque la variable explicative x_j augmente de 1 puisque

$$exp(\beta_j) = \frac{exp(\beta_0+\beta_1x_{i,1}+...+\beta_j(x_{i,j}+1)+...+\beta_px_{i,p})}{exp(\beta_0+\beta_1x_{i,1}+...+\beta_jx_{i,j}+...+\beta_px_{i,p})}$$

En inversant la fonction *logit*, on vérifie également que

$$P[Y_i=1] = \pi(x_{i,1},...,x_{i,p}) = \frac{exp(\beta_0+\beta_1x_{i,1}+...+\beta_px_{i,p})}{1+exp(\beta_0+\beta_1x_{i,1}+...+\beta_px_{i,p})}$$

La fonction logit et son inverse sont représentées graphiquement ci-dessous.

(c) **Modèle GLM Poisson.** La fonction lien canonique est la fonction *log*. Le GLM ainsi défini est le **GLM log-Poisson** qui est couramment utilisé en actuariat pour modéliser le nombre de sinistres. On suppose alors que

$$log(E[Y_i]) = \log(\lambda(x_{i,1},...,x_{i,p})) = \beta_0+\beta_1x_{i,1}+...+\beta_px_{i,p}$$

et on a donc

$$Y_i \sim\ Pois(exp(\beta_0+\beta_1x_{i,1}+...+\beta_px_{i,p})).$$

Comme $var(Y_i)=E[Y_i]$, on obtient un modèle hétéroscédastique (la variance croit avec l'espérance).

(d) **Modèle GLM gamma.** Le lien canonique est la fonction inverse $g(x)=\frac{1}{x}$. On suppose alors que

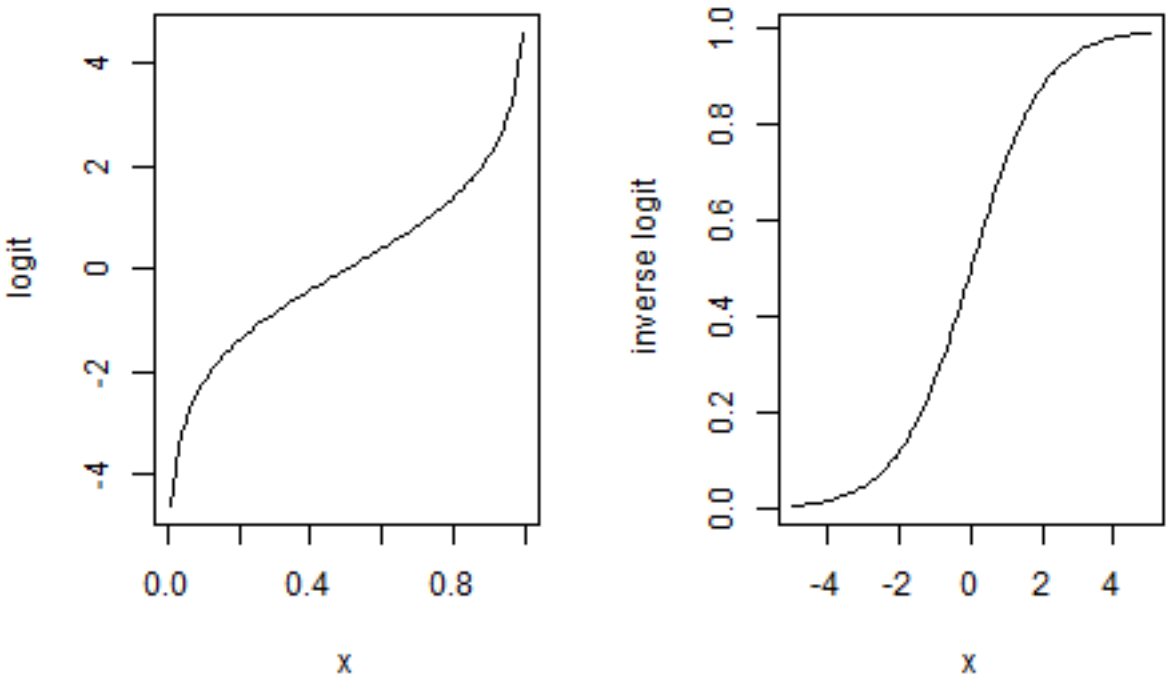
$$\frac{1}{E[Y_i]} = \frac{1}{\alpha\sigma(x_{i,1},...,x_{i,p})} = \beta_0+\beta_1x_{i,1}+...+\beta_px_{i,p}$$

En actuariat, on utilise plutôt la fonction lien *log*, notamment pour modéliser le montant des sinistres en tarification. Le GLM ainsi défini est le **GLM log-Gamma**. On a alors

$$ln(E[Y_i]) = ln(\alpha\sigma(x_{i,1},...,x_{i,p})) = \beta_0+\beta_1x_{i,1}+...+\beta_px_{i,p}.$$

On en déduit également que le modèle est hétéroscédastique puisque $var(Y_i)=\frac{E[Y_i]}{\alpha}$ dépend des variables explicatives.

```
#Représentation de la fonction logit et de son inverse
par(mfrow=c(1,2))
curve(log(x/(1-x)),0,1,ylab='logit')
curve(exp(x)/(1+exp(x)),-5,5,ylab='inverse logit')
```



En résumé, un modèle GLM est caractérisé par

1. une famille de distribution paramétrique
2. une fonction lien

qui doivent être adaptées aux données qu'on cherche à modéliser.

En tarification, une modélisation coût-fréquence est généralement utilisée. On suppose alors que le montant annuel des sinistres Y_i est donné par

$$Y_i = \sum_{k=1}^{N_i} Z_{i,k}$$

avec N_i le nombre de sinistres, $Z_{i,k}$ les montants de sinistres et la convention $Y_i = 0$ si $N_i = 0$. On modélise ensuite N_i en utilisant un modèle GLM log-Poisson et $Z_{i,k}$ par un GLM log-Gamma. Si on suppose que les variables aléatoires $(N_i, Z_{i,1}, ..., Z_{i,k}, ...)$ sont indépendantes , alors on a la formule usuelle suivante pour la prime pure

$$E[Y_i] = E[N_i]E[Z_{i,1}].$$

Sous les hypothèses ci-dessus, Y_i suit une loi Poisson-Gamma composée qui est un cas particulier de la loi de Tweedie. Une alternative au modèle cout-fréquence ci-dessus est alors d'utiliser un modèle GLM Tweedie.

4.3 Estimation des paramètres

La méthode la plus usuelle pour estimer les paramètres inconnus θ d'un GLM est la méthode du maximum de vraisemblance. D'après l'hypothèse d'indépendance, la fonction de vraisemblance est donnée par

$$L(\theta) = p(y_1, ..., y_n; \theta) = \prod_{i=1}^{n} p(y_i; \theta)$$

avec $p(y_i; \theta)$ la densité de la loi de Y_i qui dépend de la famille paramétrique choisie (Bernoulli, Gamma, Gaussien, Poisson, etc) ainsi que de la fonction lien. Par exemple :

- Pour la **régression logistique**, on a

$$p(y_i;\theta) = P[Y_i = y_i] = \begin{cases} \pi(x_{i,1},...,x_{i,p}) & \text{si } y_i = 1 \\ 1-\pi(x_{i,1},...,x_{i,p}) & \text{si } y_i = 0 \end{cases}$$

et donc

$$p(y_i;\theta) = \pi(x_{i,1},...,x_{i,p})^{y_i}(1-\pi(x_{i,1},...,x_{i,p}))^{1-y_i}$$

et la fonction de vraisemblance est donc donnée par

$$L(\theta) = \prod_{i=1}^{n} \left(\frac{exp(\beta_0+\beta_1 x_{i,1}+...+\beta_p x_{i,p})}{1+exp(\beta_0+\beta_1 x_{i,1}+...+\beta_p x_{i,p})}\right)^{y_i} \left(1-\frac{exp(\beta_0+\beta_1 x_{i,1}+...+\beta_p x_{i,p})}{1+exp(\beta_0+\beta_1 x_{i,1}+...+\beta_p x_{i,p})}\right)^{1-y_i}.$$

- Pour le **modèle linéaire gaussien**, on peut vérifier que

$$L(\theta) = \frac{1}{\sigma^n(2\pi)^{n/2}} \exp\left(-\sum_{i=1}^{n}\frac{(y_i-(\beta_0+\beta_1 x_{i,1}+...+\beta_p x_{i,p}))^2}{2\sigma^2}\right).$$

L'estimateur du maximum de vraisemblance est alors obtenu en cherchant la valeur des paramètres qui réalise le maximum de la fonction de vraisemblance

$$\hat{\theta}_n = \underset{\theta}{argmax}\{L(\theta)\}$$

A part pour le cas particulier du modèle linéaire gaussien (l'estimateur du maximum de vraisemblance de β coïncide avec l'estimateur des moindres carrés), il n'existe en général pas d'expression analytique pour $\hat{\theta}$. Des algorithmes d'optimisation numérique, usuellement de type quasi-Newton, sont alors utilisés. Comme c'est le cas pour de très nombreux modèles paramétriques en statistique, sous des conditions générales, les estimateurs du maximum de vraisemblance ont de bonnes propriétés asymptotiques. Si on note θ^* la vrai valeur des paramètres, on a

- **Consistance** : $\hat{\theta}_n \to \theta^*$ *p.s.* lorsque $n \to +\infty$
- **Normalité asymptotique** : $\sqrt{n}(\hat{\theta}_n - \theta^*) \approx \mathcal{N}(0,\Sigma)$ lorsque n est "grand" avec $\Sigma = I(\theta^*)^{-1}$ et $I(\theta^*) = -\frac{1}{n}E[l''(\theta^*)]$ la matrice d'information de Fisher (l'' désigne la matrice Hessienne de la log-vraisemblance $l(\theta) = ln(L(\theta))$.

Cette dernière propriété fournit une approximation de la loi des estimateurs pour n grand ce qui permet de faire des tests et des intervalles de confiance.

1. **Intervalles de confiance.** On suppose que pour n "grand" l'approximation suivante est valable

$$\sqrt{n}(\hat{\theta}_n(i) - \theta^*(i)) \approx \mathcal{N}(0,\sigma^2_{i,i})$$

avec $\sigma^2_{i,i}$ le ième terme diagonal de la matrice Σ, puis on utilise les quantiles de la loi normale. La fonction *R confint* permet de faire aisément les applications numériques.

2. **Tests sur la valeur des paramètres.** On utilise la même approximation que ci-dessus et le formalisme des tests. La fonction *R summary* donne les p-values des tests de l'hypothèse $H_0 : \beta_i = 0$ pour $i \in \{0,...,p\}$.

On peut aussi utiliser ces résultats pour faire des intervalles de confiance pour les prédictions ou des intervalles de prédiction.

Un exemple de régression logistique avec R. La fonction *glm* de *R* est la fonction de base pour ajuster les modèles dans R. Dans cet exemple, on cherche à prédire l'espèce d'iris à partir des autres variables présentes dans le jeu de données. Comme iris comporte trois variétés possibles, avant d'utiliser la régression logistique, on commence par se ramener à une variable binaire en regroupant deux espèces.

```
z=iris
z[,5]=0  #création d'une variable binaire qui décrit si species=virginica
z[iris$Species=='virginica',5]=1
fit=glm(Species~.,data=z,family = binomial)  #ajustement d'un modèle de régression logistique
summary(fit)  #résumé de l'ajustement
```

```
##
## Call:
## glm(formula = Species ~ ., family = binomial, data = z)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.01105  -0.00065   0.00000   0.00048   1.78065
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -42.638     25.708  -1.659   0.0972 .
## Sepal.Length   -2.465      2.394  -1.030   0.3032
## Sepal.Width    -6.681      4.480  -1.491   0.1359
## Petal.Length    9.429      4.737   1.990   0.0465 *
## Petal.Width    18.286      9.743   1.877   0.0605 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 190.954  on 149  degrees of freedom
## Residual deviance:  11.899  on 145  degrees of freedom
## AIC: 21.899
##
## Number of Fisher Scoring iterations: 12
```

```
#La qualité globale du modèle est mesurée par la déviance, cf paragraphe suivant
confint(fit)  #intervalles de confiance
```

```
## Attente de la réalisation du profilage...
```

```
##                    2.5 %     97.5 %
## (Intercept)  -118.866838 -9.8781404
## Sepal.Length   -9.099914  1.4787955
## Sepal.Width   -18.787029  0.1886707
## Petal.Length    3.332356 25.7555531
## Petal.Width     5.463643 45.7719799
```

```
predict(fit,new=z[1,],type='response') #prévision avec le modèle ajusté
```

```
##            1
## 2.220446e-16
```

```
#Ne pas oublier type='response' pour que la prévision soit à le même 'échelle' que les observations
```

4.4 Quelques remarques sur l'inférence statistique dans les GLM

Déviance. Le coefficient R^2 permet de mesurer la qualité globale d'un modèle linéaire. L'utilisation de ce critère n'est en général pas justifiée pour les modèles GLM. Dans ce cas, on utilise généralement la déviance (cf sorties de la fonction *summary.glm* de *R* ci-dessus) qui est définie par

$$D = 2(l_{sat} - \hat{l})$$

avec $\hat{l}$ la log-vraisemblance du modèle ajusté et l_{sat} la log-vraisemblance du 'modèle saturé', c'est à dire un modèle 'parfait' tel que l'espérance de la variable à prédire est égale à l'observation (c'est à dire vérifiant $E[Y_i] = y_i$). La

déviance est positive, et plus elle est proche de 0, meilleur est le modèle. L'interprétation de la valeur de la déviance est plus délicate que celle du R^2. On peut lier la différence de déviance entre deux modèles ajustés avec la statistique du test de rapport de vraisemblance décrit ci-dessous.

Test pour comparer des modèles emboîtés. Dans le paragraphe 3.1, on a vu comment on peut tester la validité d'un modèle réduit de la forme

$$H_0 : \beta_1 = 0, \beta_2 = 0, ..., \beta_q = 0 \text{ contre } H_1 : \exists i \in \{1...q\},\ \beta_i \neq 0.$$

dans le cadre du modèle linéaire gaussien en utilisant le test d'analyse de la variance (ou test de Fisher). Dans le GLM, on peut tester cette hypothèse en utilisant le test du rapport de maximum vraisemblance ou le test de Wald. La statistique du test du rapport de maximum vraisemblance est

$$X = 2(l_{H_1} - l_{H_0})$$

avec l_{H_1} le maximum de la fonction de log-vraisemblance pour le modèle complet (i.e. sans contrainte sur la valeur des coefficients) et l_{H_0} le maximum de la fonction de log-vraisemblance pour le modèle restreint (i.e. en imposant la contrainte $\beta_1 = 0, \beta_2 = 0, ..., \beta_q = 0$). Sous H_0, lorsque $n \to \infty$, X converge en loi vers une loi du χ^2 à q degrés de liberté. On accepte alors H_0 avec un risque de première espèce α si et seulement si $X \leq \chi^2_{q,1-\alpha}$.

On vérifie facilement que

$$X = 2(l_{H_1} - l_{sat}) - 2(l_{H_0} - l_{sat}) = D_{H_0} - D_{H_1}$$

est la différence entre la déviance du modèle restreint D_{H_0} et la déviance du modèle complet D_{H_1}. Le test du rapport de vraisemblance permet donc de tester la significativité de la diminution de la déviance par l'ajout de variables explicatives.

```
#Ajustement d'un modèle réduit avec une seule variable qualitative
fit0=glm(Species~Petal.Length,data=z,family = binomial)
```

```
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
```

```
anova(fit0,fit,test='LRT')  #test du rapport de maximum vraisemblance
```

```
## Analysis of Deviance Table
##
## Model 1: Species ~ Petal.Length
## Model 2: Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       148     33.432
## 2       145     11.899  3   21.533 8.157e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#H0 refusée : le modèle réduit n'est pas accepté
```

Validation de modèle. La validation du modèle linéaire gaussien repose principalement sur l'étude des résidus empiriques (éventuellement standardisés) qui doivent ressembler à la réalisation d'un échantillon gaussien i.i.d. Cette approche n'est plus valable pour les modèles GLM puisque ces modèles ne sont pas basés sur des hypothèses de normalité des résidus. Plusieurs types de résidus peuvent alors être définis pour les modèles GLM (taper *? residuals.glm* dans la console de *R*) et des graphiques spécifiques à chaque GLM/type de résidus peuvent être utilisés pour valider les hypothèses (c'est à dire la loi paramétrique et la fonction lien choisie). Ceci nécessite d'avoir bien compris la structure du modèle à valider. La fonction *R plot.lm* permet de faire des graphiques adaptés au modèle linéaire gaussien, mais il n'existe pas de fonction *plot.glm* adaptées au GLM. Les graphiques diagnostiques proposés par *R* ne sont pas pertinents pour les GLM généraux : **il ne faut donc pas chercher à interpréter les résultats donnés par la commande *R plot(fit)*.**

Sélection de modèle. Les différents outils vus dans le cadre du modèle linéaire gaussien pour faire de la sélection de modèle ou gérer les colinéarités dans les covariables , à part ceux basés sur le R^2 et le R^2_{aj}, se généralisent aux modèles GLM. On peut donc utiliser les méthodes exhaustives ou les méthodes pas à pas basées sur AIC, BIC ainsi que la validation croisée. Les méthodes Ridge et LASSO se généralisent aussi au GLM. Les codes R ci-dessous donnent un exemple de sélection de variables sur les données iris.

```
fit2=stepAIC(fit)  #sélection de modèle
```

```
## Start:  AIC=21.9
## Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

##                Df Deviance    AIC
## - Sepal.Length  1   13.266 21.266
## <none>              11.899 21.899
## - Sepal.Width   1   15.492 23.492
## - Petal.Width   1   23.772 31.772
## - Petal.Length  1   25.902 33.902

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

##
## Step:  AIC=21.27
## Species ~ Sepal.Width + Petal.Length + Petal.Width

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

##                Df Deviance    AIC
## <none>              13.266 21.266
## - Sepal.Width   1   20.564 26.564
## - Petal.Length  1   27.399 33.399
## - Petal.Width   1   31.512 37.512
```

```
summary(fit2)  #La variable Sepal.Length a été enlevée du modèle
```

```
##
## Call:
## glm(formula = Species ~ Sepal.Width + Petal.Length + Petal.Width,
##     family = binomial, data = z)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.75795  -0.00043   0.00000   0.00026   1.92193
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)   -50.527     23.995  -2.106   0.0352 *
## Sepal.Width    -8.376      4.761  -1.759   0.0785 .
## Petal.Length    7.875      3.841   2.050   0.0403 *
## Petal.Width    21.430     10.707   2.001   0.0453 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 190.954  on 149  degrees of freedom
## Residual deviance:  13.266  on 146  degrees of freedom
## AIC: 21.266
##
## Number of Fisher Scoring iterations: 12
```

```
# #Supplément : utilisation de LASSO pour la sélection de variable
# tabl=exp(seq(-9,-1,by=.1))  #Domaine pour la recherche d'un lambda optimal
# fit = glmnet(as.matrix(z[,1:4]),as.matrix(z[,5]),alpha=1,lambda=tabl,family="binomial")  #Régression LASS
# #optimisation de lambda par validation croisée
# cvfit = cv.glmnet(as.matrix(z[,1:4]),as.matrix(z[,5]),alpha=1,family="binomial",lambda=tabl,type.measure=
# #différentes mesure de performance peuvent être choisis
# #https://en.wikipedia.org/wiki/Precision_and_recall
# #par défaut, cv.glmnet utilise la déviance pour la régression logistique
# #Ici on choisit le taux de mauvaise classification
# plot(fit,xvar="lambda")
# abline(v=log(cvfit$lambda.min))
# abline(v=log(cvfit$lambda.1se))
# #ajout de la légende
# vnat=coef(fit)
# vnat=vnat[-1,ncol(vnat)]
# axis(2, at=vnat,line=-2,label=names(iris[,1:4]),las=1,tick=FALSE, cex.axis=1,col='red')
# #Toutes les variables sont conservés si on choisit le lambda qui minimise l'erreur de mauvaise classifica
# #Si on utilise la déviance, toutes les variables sont également conservées
# #représentation des coefficients
# plot(cvfit) #Score choisi en fonction de lambda
```

Quelques références

Cornillon, P. A., Matzner-Lober, E. (2007). *Régression: théorie et applications.* Springer.

Dobson, A.J., Barnett, A.G., (2018), *An Introduction to Generalized Linear Models*, Chapman and Hall/CRC; 3nd edition.

de Jong, P., Heller G. Z., (2008), *Generalized Linear Models for Insurance Data*, Cambridge University Press.

Hastie T., Tibshirani R., Friedman J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.*. Second Edition. Springer. https://web.stanford.edu/~hastie/Papers/ESLII.pdf