Sciences des données : apprentissage statistique

Pierre Barbillon, Céline Lévy-Leduc et Laure Sansonnet

2021-2022

Table des matières

Introduction générale

Le but de ce cours est de présenter des méthodes pour faire de la modélisation et de l'analyse de données. Il s'inscrit donc pleinement dans le domaine de la **science des données** dont le but est d'extraire de la connaissance à partir de données en utilisant des méthodes venant des mathématiques appliquées et plus particulèrement de l'apprentissage statistique, de l'informatique ou de la physique.

Plus précisément, l'**apprentissage statistique** est une branche récente des statistiques qui consiste à mettre en place des outils pour modéliser et comprendre des données complexes. C'est une discipline qui mélange des compétences en mathématiques appliquées (statistique et optimisation notamment) et en informatique (*machine learning* en anglais que l'on traduit usuellement par *apprentissage automatique*). Avec l'explosion des "Big Data" ou "données massives", l'apprentissage statistique est devenu ces dernières années une discipline importante à la fois au sein du monde académique et industriel.

L'apprentissage statistique se subdivise essentiellement en deux catégories :

- l'apprentissage supervisé (voir Partie II),
- l'apprentissage non supervisé (voir Partie I).

On dit que l'on se trouve dans la **première situation (apprentissage supervisé)** lorsque l'on a une variable réponse que l'on souhaite expliquer par d'autres variables que l'on appelle également des *prédicteurs*. Dans ce cas, on souhaite trouver la fonction f qui permettra de relier la réponse y aux prédicteurs à un terme d'erreur ε près. Plus précisément, une représentation classique et généralement utilisée consiste à estimer f telle que :

$$y = f(x_1, x_2, \ldots, x_p) + \varepsilon$$

à partir d'observations $(y_i)_{1 \leq i \leq n}$ que l'on modélise comme des réalisations de variables aléatoires Y_i définies par :

$$Y_i = f(x_{i,1}, x_{i,2}, \ldots, x_{i,p}) + E_i, \;\; 1 \leq i \leq n,$$

où E_i est une variable aléatoire d'espérance nulle et de variance σ^2 modélisant l'erreur de mesure et/ou de modélisation associée à l'observation i et $x_{i,1}, x_{i,2}, \ldots, x_{i,p}$ sont les valeurs des p prédicteurs pour l'observation i. L'intérêt de l'apprentissage supervisé est de pouvoir faire de la **prédiction** ou de l'**inférence**. En effet, si l'on sait estimer f par $\widehat{f}$, on pourra **prédire** la valeur de Y_{n+1} uniquement à partir des prédicteurs $x_{n+1,1}, x_{n+1,2}, \ldots, x_{n+1,p}$ par

$$\widehat{Y}_{n+1} = \widehat{f}(x_{n+1,1}, x_{n+1,2}, \ldots, x_{n+1,p}).$$

L'**inférence**, quant à elle, permet de comprendre l'influence et l'importance de chacun des p prédicteurs sur la variable réponse y. En particulier, il sera possible d'utiliser des règles de décision telles que des tests statistiques pour mettre en évidence ou non l'effet d'un prédicteur particulier x_k sur y.

L'apprentissage supervisé est un domaine vaste. Nous présenterons dans ce cours uniquement :

- les modèles linéaires qui supposent une forme particulière pour la fonction f : il s'agit d'une combinaison linéaire des prédicteurs (voir Chapitres 3 et 4),
- la régression logistique (voir Chapitre 5),

- la méthode des K plus proches voisins (voir Chapitre 6).

Notons que les deux premières méthodes permettent des interprétations simples de l'effet des prédicteurs sur la réponse y tandis que la dernière a pour finalité principale la prédiction.

On dit que l'on se trouve dans la **seconde situation (apprentissage non supervisé)** lorsque l'on dispose de plusieurs variables (prédicteurs) observées sur différents individus mais que l'on n'a pas de variable réponse associée à expliquer. On dit que l'on travaille en "non supervisé" parce que l'on ne dispose pas de variable réponse pour "superviser" l'analyse. Dans ce cas, les données se présentent sous forme d'un tableau ayant la forme suivante :

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

où $x_{i,k}$ correspond à la valeur de la variable k pour l'individu i (par exemple, la colonne 1 du tableau correspond à la valeur prise par les n individus de l'étude pour la 1ère variable et la ligne 2 correspond aux valeurs des p variables pour l'individu 2 de l'étude).

Dans ces cas-là différentes approches peuvent être considérées :

- une approche de réduction de dimension permettant de visualiser le tableau de données en utilisant de nouvelles variables construites à partir des variables de départ (Analyse en Composantes Principales, voir Chapitre 1)
- une approche de *classification* (ou *clustering* en anglais) qui consiste à créer des groupes d'individus ayant des caractéristiques similaires à partir des valeurs des variables (k-means et Classification Ascendante Hiérarchique, voir Chapitre 2)

Il est important de noter que dans les méthodes d'apprentissage supervisé présentées, il y a un modèle probabiliste sous-jacent alors que dans les méthodes d'apprentissage non supervisé, il n'y en a pas.

<u>Références :</u>

— **T. Hastie, R. Tibshirani and J. Friedman.** The Elements of Statistical Learning. Springer, 2008.
— **G. James, D. Witten, T. Hastie and R. Tibshirani.** An Introduction to Statistical Learning *with applications in R*. Springer, 2013.
— **J.-J. Daudin** (coordination)**.** Le modèle linéaire et ses extensions. Ellipses, 2015.

Première partie

Apprentissage non supervisé : visualisation de données

Chapitre 1

Méthode de réduction de dimension : analyse en composantes principales

1.1 Rappels d'algébre linéaire et bilinéaire

Nous considérons des vecteurs pouvant être notés en lettres capitales (par exemple U) qui appartiennent à un espace vectoriel $\mathbb{R}^p$, c'est-à-dire qu'un tel vecteur U a p coordonnées :

$$U = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_p \end{pmatrix}.$$

Par convention, il est écrit en colonne. Les coordonnées correspondent à un choix de base de l'espace $\mathbb{R}^p$.

Définition 1 (Base d'un espace vectoriel)**.** *Une base d'un espace vectoriel est un ensemble de vecteurs formant une famille libre et génératrice de cet espace vectoriel. Autrement dit, une base permet d'écrire de manière unique tout vecteur appartenant à l'espace vectoriel.*

La base canonique est un exemple d'une telle base, elle contient les vecteurs $E_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$, $E_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ldots, E_p = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$. Dans cette base, le vecteur U s'écrit comme une combinaison

linéaire des vecteurs de base :

$$U = u_1 E_1 + u_2 E_2 + \ldots + u_p E_p \,.$$

Définition 2 (Dimension d'un espace vectoriel)**.** *La dimension d'un espace vectoriel correspond à son nombre de vecteurs de base.*

Ainsi $\mathbb{R}^p$ est un espace vectoriel de dimension p.

On peut définir sur $\mathbb{R}^p$ un produit scalaire qui permettra de définir une norme et la notion d'orthogonalité.

Définition 3 (Produit scalaire et Norme)**.** *Le produit scalaire sur $\mathbb{R}^p$ est une fonction de $\mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ définie et notée ainsi : pour $U \in \mathbb{R}^p$ et $V \in \mathbb{R}^p$:*

$$U.V = U^T V = \sum_{j=1}^{p} u_j v_j \,,$$

où U^T est la transposée du vecteur U c'est-à-dire le vecteur ligne $(u_1, u_2, \ldots, u_p)$ et $U^T V$ est un calcul matriciel (un vecteur ligne multiplié par un vecteur colonne) qui donne un scalaire (un nombre réel).
La norme d'un vecteur $U \in \mathbb{R}^p$ est notée $\|U\|$ et est définie comme la racine carrée du produit scalaire $U.U$, ainsi

$$\|U\| = \sqrt{U.U} = \sqrt{\sum_{j=1}^{p} u_j^2} \,.$$

La norme permet de définir une distance entre 2 vecteurs U et V de $\mathbb{R}^p$ comme la norme du vecteur $(U - V)$. On notera alors

$$d(U, V) = \|U - V\| \,.$$

La projection orthogonale d'un vecteur $U \in \mathbb{R}^p$ sur une droite Δ s'obtient alors grâce au produit scalaire.

Proposition 1. *Pour $A \in \mathbb{R}^p$ un vecteur directeur unitaire (de norme 1 : $\|A\| = 1$) de Δ, la projection orthogonale de $U \in \mathbb{R}^p$ sur Δ est $(U.A)\ A$.*

On a alors orthogonalité entre 2 vecteurs lorsque la projection de l'un sur l'autre donne le vecteur nul. Nous donnons ainsi ci-dessous les définitions de vecteurs orthogonaux, d'une base orthonormée et d'une matrice orthogonale.

Définition 4 (Vecteurs orthogonaux, Base orthonormée et Matrice orthogonale)**.**
Les vecteurs $U, V \in \mathbb{R}^p$ sont orthogonaux si $U.V = 0$.
La base des vecteurs $(U_1, U_2, \ldots, U_p)$ est orthonormée si tous les vecteurs de base sont orthogonaux 2 à 2 ($U_i.U_{i'} = 0$ pour $i \neq i'$) et si tous les vecteurs sont unitaires ($\|U_i\| = 1$ pour tout i).
Une matrice carrée $P \in \mathbb{R}^{p \times p}$ est orthogonale si les p vecteurs colonnes de P forment une base orthonormée.

Nous pouvons remarquer que la base canonique définie ci-dessus formée des vecteurs E_1, E_2, $\ldots$, E_p est une base orthonormée. Nous avons également que si une matrice P est orthogonale on a $P^TP = Id$ où Id est la matrice identité de taille $p \times p$ c'est-à-dire la matrice avec des 1 sur la diagonale et des 0 partout ailleurs. Cela implique que $P^{-1} = P^T$. Nous rappelons que P^T est la notation pour la transposée de la matrice P c'est à dire la matrice P où l'on a inversé lignes et colonnes.

Nous utilisons la notion de matrice orthogonale dans un théorème de diagonalisation.

Théorème 1. *Soit M une matrice symétrique réelle (c'est-à-dire une matrice $M \in \mathbb{R}^{p\times p}$ telle que $M^T = M$), on peut diagonaliser M dans une base orthonormée c'est-à-dire que l'on a l'égalité*

$$M = PDP^T\,,$$

où D est une matrice diagonale :

$$D = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_p \end{pmatrix}$$

qui contient les valeurs propres $\lambda_1, \ldots, \lambda_p$ de M et P est une matrice orthogonale formée par les vecteurs propres de M rangés en colonne. P est alors la matrice de changement de base permettant de passer de la base des vecteurs propres de M à la base canonique et P^T est la matrice de changement de base permettant de passer de la base canonique à la base des vecteurs propres de M.

Nous avons encore besoin de la notion de sous-espace vectoriel orthogonal.

Définition 5 (Sous-espace orthogonal)**.** *Pour $\mathcal{F}$ un sous-espace vectoriel de $\mathbb{R}^p$, on note $\mathcal{F}^\perp$ le sous-espace vectoriel de $\mathbb{R}^p$ contenant tous les vecteurs orthogonaux à ceux de $\mathcal{F}$. Formellement, on a*

$$\mathcal{F}^\perp = \{U \in \mathbb{R}^p \mid \forall V \in \mathcal{F},\ U.V = 0\}\ .$$

Le sous-espace orthogonal $\mathcal{F}^\perp$ est un sous-espace supplémentaire à $\mathcal{F}$, c'est-à-dire que tout vecteur $W \in \mathbb{R}^p$ peut s'écrire $W = U + V$ avec $U \in \mathcal{F}$ et $V \in \mathcal{F}^\perp$. On note alors

$$\mathbb{R}^p = \mathcal{F} \overset{\perp}{\oplus} \mathcal{F}^\perp\,.$$

On a l'égalité en termes de dimension $p = \dim(\mathcal{F}) + \dim(\mathcal{F}^\perp)$. Plus concrètement, dans l'espace classique à 3 dimensions $\mathbb{R}^3$, si $\mathcal{F}$ correspond à un axe (c'est-à-dire une droite de dimension 1), le sous-espace orthogonal est le plan (de dimension 2) tel que l'axe soit dirigé par un vecteur orthogonal au plan.

1.2 Type de données et motivations

Nous supposons que nous disposons de données concernant n individus pour lesquels nous avons p variables **quantitatives** les décrivant. Les données sont organisées sous la forme d'une matrice $n \times p$

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,p} \\ x_{2,1} & x_{2,2} & \dots & x_{2,p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \dots & x_{n,p} \end{pmatrix} \tag{1.1}$$

avec les individus en ligne et les variables en colonne. Nous notons alors U_i $(1 \leq i \leq n)$ les vecteurs lignes, on a donc $U_i \in \mathbb{R}^p$ les données correspondant aux individus et nous notons V_j $(1 \leq j \leq p)$ les vecteurs colonnes, on a donc $V_j \in \mathbb{R}^n$ les données correspondant aux variables.

Une représentation classique de données pour lesquelles nous disposons de variables quantitatives est de faire un nuage de point dans un plan (donc en dimension 2, une variable en abscisse et une autre en ordonnée) voire en dimension 3. Si $p > 3$, il s'avère impossible de représenter entièrement les données. On doit choisir un sous-ensemble de 2 ou 3 variables pour avoir une représentation sous la forme d'un nuage de point. Le nombre de plans est alors de 2 parmi p ce qui donne pour par exemple $p = 11$, $p \times (p-1)/2 = 55$ graphiques à analyser. Étant donné qu'il peut y avoir des dépendances entre les p variables, il n'est sans doute pas nécessaire de faire toutes les représentations.

Le principe de l'ACP va être d'exploiter la dépendance entre les variables pour essayer de fournir une représentation résumée tenant compte de l'ensemble des variables dans un plan (ou éventuellement plusieurs). Cette représentation résumée permettra de décrire nos données et en particulier de repérer des individus marginaux (très différents des autres) ou des groupes d'individus. Puisque l'ACP permet une description des données tenant compte de l'ensemble des variables, on parlera d'une méthode de **statistique descriptive multivariée**.

Afin de se prémunir contre l'impact des choix d'unités entre les différentes variables, nous supposons pour la suite que **les données sont centrées-réduites**. C'est-à-dire que chaque variable V_j a une moyenne nulle :

$$\bar{V_j} = x_{\bullet j} = \frac{1}{n} \sum_{i=1}^{n} x_{ij} = 0$$

et

$$\mathrm{Var}(V_j) = \frac{1}{n} \sum_{i=1}^{n} (x_{ij} - x_{\bullet j})^2 = 1\,.$$

La notation $x_{\bullet j}$ signifie que l'on moyenne sur l'indice i remplacé par $\bullet$. Si cette réduction n'était pas effectuée, on pourrait artificiellement augmenter l'influence d'une variable en l'exprimant dans une unité plus faible. Par exemple, une mesure exprimée en mètre aurait sa variance multipliée par 10000 si on changeait l'unité du mètre au centimètre. Puisque les variables peuvent être de nature différente (on peut avoir des mesures en mètre et des masses en kg), il n'y a pas d'unité commune. La réduction permet alors d'harmoniser en imposant une variance de 1 pour chaque variable.

1.3 Recherche des composantes principales

Afin de déterminer si un résumé des données est meilleur qu'un autre, nous avons besoin d'un critère objectif quantifiable que nous chercherons à optimiser. Dans l'espace $\mathbb{R}^p$, il y a une

certaine dispersion dans les données. Notre but sera de trouver une projection des données dans un espace de dimension réduite qui permettra de conserver le maximum de cette dispersion. Nous définissons donc l'inertie comme mesure de dispersion des données.

Définition 6 (Inertie d'un nuage de point)**.** *Pour les données* $\boldsymbol{X}$*, nous définissons l'inertie totale du nuage de point comme*

$$I = \sum_{j=1}^{p} Var\ V_j = \sum_{j=1}^{p} \left(\frac{1}{n} \sum_{i=1}^{n} (x_{ij} - x_{\bullet j})^2 \right) = \frac{1}{n} \sum_{i=1}^{n} \|U_i - U_G\|^2 .$$

Puisque les données sont centrées réduites, nous avons $x_{\bullet j} = 0$, $U_G = (0, \ldots, 0)$ et surtout Var $V_j = 1$ pour tout j ce qui implique que $\boldsymbol{I = p}$. Ainsi la contribution à l'inertie totale est la même pour chacune des variables et est égale à 1.

Si nous considérons une projection des points sur un axe Δ une partie de la dispersion totale des données (inertie) sera perdue. Nous notons les projections orthogonales des individus U_i sur l'axe Δ : $h_\Delta(U_i)$ et on pose la définition suivante.

Définition 7 (Inertie perdue par projection)**.** *L'inertie perdue en projetant orthogonalement les individus sur un axe* Δ *est*

$$I_\Delta = \frac{1}{n} \sum_{i=1}^{n} dist(U_i, \Delta)^2 = \frac{1}{n} \sum_{i=1}^{n} \|U_i - h_\Delta(U_i)\|^2 .$$

Naturellement, cette définition s'étend à des sous-espaces vectoriels de dimension quelconque, les axes étant des sous-espaces vectoriels de dimension 1. Pour V un sous-espace vectoriel de $\mathbb{R}^p$, on a

$$I_V = \frac{1}{n} \sum_{i=1}^{n} \text{dist}(U_i, V)^2 = \frac{1}{n} \sum_{i=1}^{n} \|U_i - h_V(U_i)\|^2 ,$$

où les $h_V(U_i)$ sont les projections des U_i sur V.

Le théorème de Huygens est assez simple à démontrer et donne une décomposition de l'inertie entre la projection sur un sous-espace vectoriel et son supplémentaire orthogonal.

Théorème 2 (Théorème de Huygens)**.** *Pour* V *un sous-espace vectoriel de* $\mathbb{R}^p$*, la décomposition de l'inertie est :*

$$I = I_V + I_{V^\perp}$$

où $V^\perp$ *est l'espace orthogonal à* V*.*

Preuve du théorème de Huygens La preuve repose simplement sur l'application du théorème de Pythagore dans l'espace $\mathbb{R}^p$. La figure 1.1 est dans le plan donc le sous-espace considéré est un axe Δ donc $V = \Delta$ et $V^\perp$ est également un axe. Par le théorème de Pythagore appliqué sur le triangle inférieur de la figure, nous avons

$$\|U_i\|^2 = \|U_i - h_V(U_i)\|^2 + \|h_V(U_i)\|^2 .$$

On remarque sur la figure 1.1 que $\|h_V(U_i)\| = \|U_i - h_{V^\perp}(U_i)\|$ (les vecteurs surlignés en bleu) où $h_{V^\perp}(U_i)$ est la projection orthogonale de U_i sur $V^\perp$.

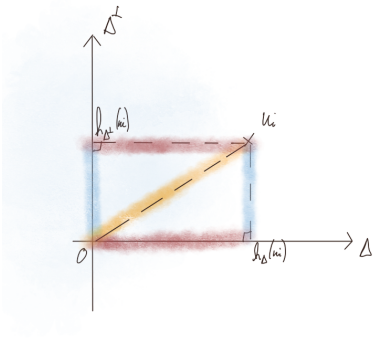


FIGURE 1.1 – Projection de l'individu U_i sur l'axe Δ et sur le sous-espace vectoriel orthogonal à Δ. Les vecteurs surlignés de la même couleur ont la même norme.

Ainsi

$$\|U_i\|^2 = \|U_i - h_V(U_i)\|^2 + \|U_i - h_{V^\perp}(U_i)\|^2 .$$

Cette égalité étant valable pour tout i, il suffit de sommer ces égalités sur i et de diviser par n pour obtenir le résultat du théorème. □

L'inertie $I_{V^\perp}$ est par définition l'inertie perdue en projetant sur $V^\perp$ mais par le théorème de Huygens nous obtenons qu'elle correspond aussi à **l'inertie conservée en projetant sur** Δ. En effet, le théorème de Huygens montre que $I_{V^\perp}$ est le complémentaire à l'inertie totale de l'inertie perdue en projetant sur Δ : I_V. Cette interprétation est renforcée par son expression obtenue lors de la preuve ci-dessus :

$$I_{V^\perp} = \frac{1}{n}\sum_{i=1}^{n} \|h_V(U_i)\|^2 .$$

Nous chercherons donc des sous-espaces sur lesquels la projection des individus permettra de conserver le maximum d'inertie ce qui est équivalent à minimiser l'inertie perdue. Ces sous-espaces seront déterminés grâce à l'identification d'axes principaux (appelées également composantes principales). On cherche donc p axes notés $\Delta_1, \Delta_2, \ldots, \Delta_p$ orthogonaux les uns aux autres donnant une décomposition de $\mathbb{R}^p$:

$$\mathbb{R}^p = \Delta_1 \overset{\perp}{\oplus} \Delta_2 \overset{\perp}{\oplus} \cdots \overset{\perp}{\oplus} \Delta_p$$

et d'importance décroissante c'est-à-dire tels que l'inertie conservée soit maximale sur Δ_1 puis maximale sur Δ_2 orthogonalement à Δ_1 et ainsi de suite. On a $I_{\Delta_1^\perp} \geq I_{\Delta_2^\perp} \geq \cdots \geq I_{\Delta_p^\perp}$ avec $I_{\Delta_1^\perp} + I_{\Delta_2^\perp} + \cdots + I_{\Delta_p^\perp} = I = p$. L'ACP sera utile sur un jeu de données si on peut obtenir une part d'inertie conservée sur les premiers axes, par exemple la part d'inertie conservée sur le premier plan : $(I_{\Delta_1^\perp} + I_{\Delta_2^\perp})/I$, proche de 100%.

Le théorème suivant nous donne les axes principaux et les inerties conservées associées :

Théorème 3 (Composantes principales)**.** *Pour Σ la matrice de taille $p \times p$ des corrélations des variables V_j ($\Sigma = \frac{1}{n}\sum_{i=1}^{n} U_i U_i^T$ car les données sont centrées-réduites), les composantes principales sont dirigées par les vecteurs propres de Σ associés aux valeurs propres ordonnées par ordre décroissant. Plus explicitement, on a la diagonalisation de Σ dans la base orthonormée de ses vecteurs propres : $\Sigma = PDP^T$ où P est une matrice orthogonale composée des vecteurs propres rangés en colonne et*

$$D = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_p \end{pmatrix},$$

avec $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$. La part d'inertie conservée sur chaque composante principale est $I_{\Delta_k^\perp} = \lambda_k$ pout tout $1 \leq k \leq p$.

Preuve du théorème.

Commençons par chercher le premier axe qui doit maximiser $I_{\Delta^\perp}$, rappelons que

$$I_{\Delta^\perp} = \frac{1}{n}\sum_{i=1}^{n} \|h_\Delta(U_i)\|^2 .$$

Pour $a \in \mathbb{R}^p$ un vecteur directeur de Δ tel que $\|a\| = 1$, on a $\|h_\Delta(U_i)\|^2 = (U_i.a)^2 = a^T U_i U_i^T a$ d'où

$$\begin{aligned} I_{\Delta^\perp} &= a^T \left(\frac{1}{n}\sum_{i=1}^{n} U_i U_i^T\right) a = a^T \Sigma a \\ &= a^T P D P^T a = b^T D b = \sum_{j=1}^{p} \lambda_j b_j^2 . \end{aligned}$$

où on a posé $b = P^T a$. Remarquons que $\|b\| = \sqrt{b^T b} = \sqrt{b^T b} = \sqrt{a^T P P^T a} = \|a\|$ car $PP^T = Id$ car P est une matrice orthogonale. Ainsi b reste un vecteur orthogonal dont les coordonnées sont exprimées dans la base des vecteurs propres de Σ. Il est clair que maximiser $\sum_{j=1}^{p} \lambda_j b_j^2$ sous la contrainte $\|b\| = 1$ qui est équivalente à $\sum_{j=1}^{p} b_j^2 = 1$ revient à choisir $b = (1, 0, \ldots, 0)$ puisque $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$. Nous avons donc montré que Δ_1 est dirigé par le vecteur propre de Σ associé à la valeur propre λ_1 et que l'inertie conservée sur cet axe est $I_{\Delta_1^\perp} = \lambda_1$.

Pour les axes suivants, nous cherchons toujours à maximiser l'inertie conservée sous la condition d'être orthogonal aux premiers axes. Ainsi le deuxième axe est cherché dans l'espace orthogonal à Δ_1, ce sous-espace admet pour base orthormée les $p-1$ vecteurs propres restant. Le deuxième axe Δ_2 est alors dirigé par le vecteur propre de Σ associé à la valeur propre λ_2 et l'inertie conservée sur cet axe est $I_{\Delta_2^\perp} = \lambda_2$. Et ainsi de suite.... □

Le théorème ci-dessus nous indique donc comment trouver les axes principaux (en diagonalisant la matrice des corrélations) et nous donne la dispersion conservée sur chacun des axes. Le

but étant de résumer les données par moins de dimensions que les p dimensions initiales, on pourra choisir de conserver les k premiers axes tels que

$$(I_{\Delta_1^\perp} + I_{\Delta_2^\perp} + \ldots + I_{\Delta_k^\perp})/I = (\lambda_1 + \lambda_2 + \ldots + \lambda_k)/p$$

dépasse un certain pourcentage (par exemple 90%). C'est bien une réduction de dimension car on étudiera nos données sur k dimensions (idéalement peu 2 ou 3) plutôt que les p dimensions initiales.

Remarque 1. *La recherche des composantes principales peut également s'interpréter comme la recherche de nouvelles variables $Z_1, Z_2, \ldots, Z_p$ combinaisons linéaires des anciennes variables, non corrélées entre elles et telles que la variance de Z_1 soit maximale, puis la variance de Z_2, etc. Commençons par chercher Z_1 comme suit :*

$$Z_{i,1} = a_1 x_i + a_2 x_{i,2} + \cdots + a_p x_{i,p} = \sum_{j=1}^{p} a_j x_{i,j},\ 1 \leq i \leq n,$$

telle qu'elle ait la plus grande variance empirique possible sous la contrainte $\sum_{j=1}^{p} a_j^2 = 1$. Sans cette normalisation il suffirait de prendre les a_j les plus grands possibles pour avoir la plus grande variance empirique possible. Autrement dit, on cherche à résoudre le problème suivant :

$$\max_{a_1,\ldots,a_p} Var(Z_1) \text{ sous la contrainte } \sum_{i=1}^{p} a_i^2 = 1.$$

On cherche donc à résoudre :

$$\max_{a_1,\ldots,a_p} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(Z_{i,1} - \overline{Z_1} \right)^2 \right\} \text{ sous la contrainte } \sum_{i=1}^{p} a_i^2 = 1,$$

où $\overline{Z_1} = \sum_{j=1}^{p} a_j x_{\bullet,j}$ avec $x_{\bullet,j} = 0$, soit encore

$$\max_{a} \left\{ \frac{1}{n} \sum_{i=1}^{n} (a^T U_i)^2 \right\} \text{ sous la contrainte } \|a\|^2 = 1.$$

Or $a^T \left(\frac{1}{n} \sum_{i=1}^{n} U_i U_i^T \right) a = \frac{1}{n} \sum_{i=1}^{n} (a_1^T U_i)^2$, on fait alors face au même problème d'optimisation que précédemment. Ainsi a donnant les coefficients de la combinaison linéaire définissant Z_1 est bien le premier vecteur propre de la matrice des corrélations Σ. Pour les variables suivantes, on constate que la condition de corrélation nulle entre les nouvelles variables $Z_1, Z_2, \ldots, Z_p$ implique que les vecteurs contenant les coefficients des combinaisons linéaires doivent être orthogonaux. En effet, pour $Z_1 = a_1^T U_i$ et $Z_2 = a_2^T U_i$, on a

$$Cov(Z_2, Z_1) = \frac{1}{n} \sum_{i=1}^{n} \left(Z_{i,1} - \overline{Z_1} \right) \left(Z_{i,2} - \overline{Z_2} \right) = \frac{1}{n} \sum_{i=1}^{n} a_2' U_i U_i' a_1 = a_2' \Sigma a_1 = \lambda_1 a_2' a_1,$$

car a_1 est un vecteur propre de Σ. D'où l'on déduit que $Cov(Z_2, Z_1) = 0 = a_2' a_1$ ce qui signifie que les vecteurs a_1 et a_2 sont orthogonaux. Cette recherche de la seconde composante principale mènera bien au second vecteur propre de Σ et ainsi de suite.

1.4 Représentation des variables et des individus

Une fois les axes identifiés, nous avons donc des axes que nous pouvons considérer comme étant de nouvelles variables. On note $Z_k \in \mathbb{R}^n$ une nouvelle variable (correspondant à l'axe Δ_k). Ces variables sont des combinaisons linéaires des anciennes variables V_j :

$$Z_k = \sum_{j=1}^{p} p_{jk} V_j$$

où les p_{jk} sont les éléments de la matrice P.

Ces nouvelles variables ne sont pas corrélées entre elles et ont des variances correspondant aux inerties conservées sur les axes : $\mathrm{Var}(Z_k) = \lambda_k$.

Afin de relier ces nouvelles variables difficilement interprétables aux anciennes ayant un sens pour nos données, nous calculons les corrélations données dans la proposition suivante.

Proposition 2. *Les corrélations sont :*

$$Cor(Z_k, V_j) = \sqrt{\lambda_k} p_{jk} \,.$$

En sommant sur toutes les nouvelles variables on obtient

$$\sum_{k=1}^{p} Cor(Z_k, V_j)^2 = 1 \,.$$

Une représentation classique de ces corrélations se lit sur le cercle des corrélations pour un plan choisi (formé de deux axes). On représente les anciennes variables par des flèches dont la pointe a pour abscisse la corrélation de l'ancienne variable avec la nouvelle variable correspondant à l'axe choisi à l'horizontale et pour ordonnée la corrélation de l'ancienne variable avec la nouvelle variable correspondant à l'axe choisi à la verticale. Typiquement, la figure 1.2 représente le cercle où l'axe horizontal est le premier axe Δ_1 et l'axe à la verticale est le deuxième axe Δ_2. Les variables bien représentées sur le plan correspondent aux flèches proches du cercle. Si la flèche est courte c'est que l'ancienne variables V_j est corrélée à d'autres nouvelles variables (ou dit autrement, mieux représentée sur d'autres axes) d'après la deuxième partie de la proposition ci-dessus. S'intéresser aux angles entres les anciennes variables bien représentées (flèches proches du cercle) permet de retrouver les valeurs des corrélations entre elles comme étant le cosinus de cet angle. Ceci est illustré sur la figure 1.3.

La représentation des individus se fait par rapport aux coordonnées des individus sur les premières composantes de l'ACP. Les nouvelles coordonnées des individus sur les composantes principales sont données par le vecteur $Y_i \in \mathbb{R}^p$ correspondant à l'individu U_i. Ce vecteur se calcule ainsi :

$$Y_i = P^T U_i$$

où P est la matrice des vecteurs propres de Σ obtenue lors de sa diagonalisation. Ainsi pour représenter les individus dans le premier plan principal, on représentera les n individus par rapport aux 2 premières coordonnées des Y_i. Cela correspond donc à la projection dans le plan engendré par les 2 premières composantes : $\Delta_1 \overset{\perp}{\oplus} \Delta_2$.

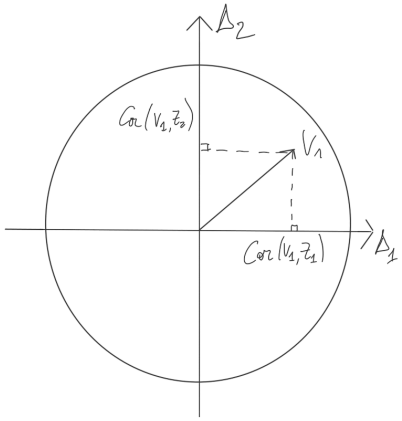


FIGURE 1.2 – Représentation d'une ancienne variable V_1 sur le cercle des corrélations correspondant aux axes Δ_1 et Δ_2.

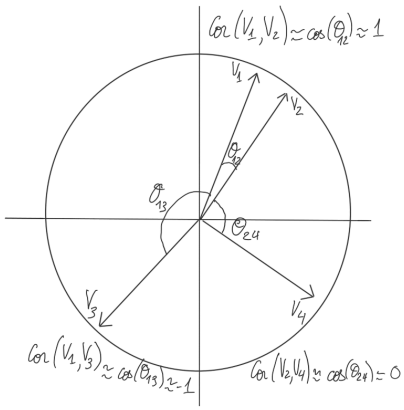


FIGURE 1.3 – Représentation des anciennes variables V_1 sur le cercle des corrélations correspondant aux axes Δ_1 et Δ_2. Pour les flèches proches du cercle, le cosinus de l'angle entre deux flèches permet de retrouver la corrélation entre les deux anciennes variables correspondantes.

Sur ce premier plan, on pourra interpréter les proximités entre les projections des individus comme étant des proximités entre les individus à condition que les projections soient proches des individus dans l'espace $\mathbb{R}^p$. Commençons par définir la qualité de représentation par rapport à un axe :

Définition 8 (Qualité de représentation d'un individu)**.** *La qualité de représentation de l'individu U_i par sa projection sur l'axe Δ_k est calculée ainsi :*

$$\cos^2\left(angle(U_i, \Delta_k)\right).$$

Cette mesure est dans $[0,1]$, elle vaut 1 si $U_i = h_{\Delta_k}(U_i)$ c'est-à-dire si l'individu est sur l'axe et vaut 0 si la projection $h_{\Delta_k}(U_i) = 0$ c'est-à-dire que l'axe Δ_k est orthogonal à U_i.

Afin d'illustrer la pertinence de cette mesure, nous pouvons nous intéresser à la figure 1.4. Si les angles sont faibles (cosinus proche de 1), il est quasiment équivalent de comparer des proximités entre les projections sur l'axe et les points dans l'espace tandis que si les angles sont grands, les proximités entre projection peuvent nous induire en erreur.

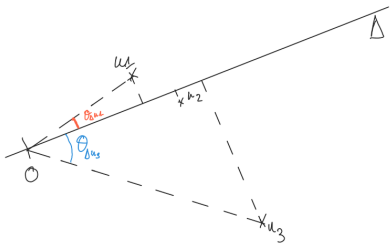


FIGURE 1.4 – Illustration de la qualité de représentation d'un individu par sa projection sur Δ en fonction du cosinus de son angle avec cet axe.

Cette mesure de qualité de représentation s'étend naturellement au plan $\Delta_1 \overset{\perp}{\oplus} \Delta_2$ par la formule :

$$\cos^2\left(\text{angle}(U_i, \Delta_1 \overset{\perp}{\oplus} \Delta_2)\right) = \cos^2\left(\text{angle}(U_i, \Delta_1)\right) + \cos^2\left(\text{angle}(U_i, \Delta_2)\right).$$

Nous pourrons alors ajouter cette information sur la représentation dans le premier plan comme une couleur ou liée à la taille du point.

1.5 Application avec R : Budget de l'état

Nous chargeons les packages `R` nécessaires ainsi que les données du budget de l'état.

```
rm(list=ls())
library(knitr) #pour avoir un format table dans les sorties
library(ggplot2) #pour avoir de 'beaux' graphiques
```

```
library(gridExtra)
library(FactoMineR) #pour effectuer l'ACP
library(factoextra) #pour extraire et visualiser les résultats issus de FactoMineR
library(corrplot) #pour avoir une représentation des corrélations
Budgets <- read.table("Etat.txt",sep="", header=T, row.names=1)
#row.names=1 signifie que la première colonne est prise comme nom de lignes
```

Il s'agit d'une table de données de dimension

```
dim(Budgets)
```

```
## [1] 24 11
```

Chaque ligne correspond à une année pour laquelle on a le pourcentage du budget de l'état dépensé pour un des 11 types de dépenses possibles. Ces 11 types de dépenses possibles correspondent aux variables. Les valeurs sont données en pourcentage du budget global et les différents types sont :

- ⋆ PVP : Pouvoirs publics,
- ⋆ AGR : Agriculture,
- ⋆ CMI : Commerce et industrie,
- ⋆ TRA : Travail,
- ⋆ LOG : Logement,
- ⋆ EDU : Education,
- ⋆ ACS : Action sociale,
- ⋆ ANC : Anciens combattants,
- ⋆ DEF : Défense,
- ⋆ DET : Remboursement de la dette, et
- ⋆ DIV : Divers.

Voici un extrait des données :

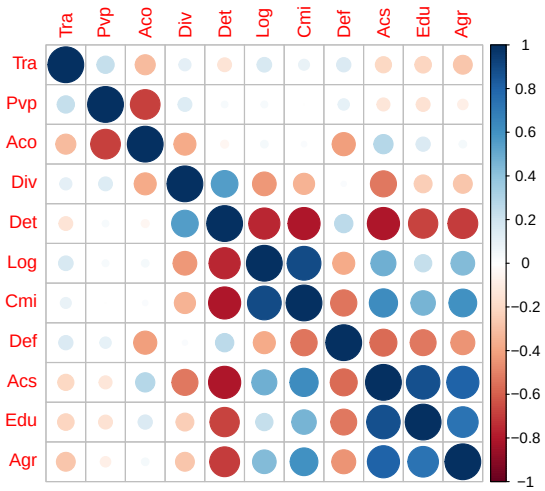
```
kable(head(Budgets))
```

	Pvp	Agr	Cmi	Tra	Log	Edu	Acs	Aco	Def	Det	Div
1872	18.0	0.5	0.1	6.7	0.5	2.1	2.0	0	26.4	41.5	2.1
1880	14.1	0.8	0.1	15.3	1.9	3.7	0.5	0	29.8	31.3	2.5
1890	13.6	0.7	0.7	6.8	0.6	7.1	0.7	0	33.8	34.4	1.7
1900	14.3	1.7	1.7	6.9	1.2	7.4	0.8	0	37.7	26.2	2.2
1903	10.3	1.5	0.4	9.3	0.6	8.5	0.9	0	38.4	27.2	3.0
1906	13.4	1.4	0.5	8.1	0.7	8.6	1.8	0	38.5	25.3	1.9

On peut représenter de manière graphique les corrélations entre les variables. On les réorganise pour trouver des blocs de variables corrélées.

```
correlation=cor(Budgets)
h=heatmap(abs(correlation),symm=T)
```

```
corrplot(correlation[h$rowInd,h$colInd])
```



On réalise à présent l'ACP.

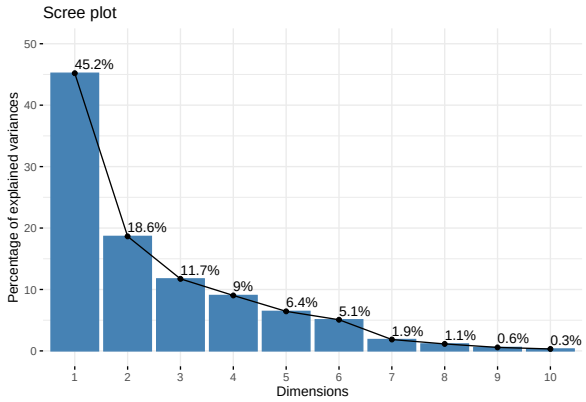
```
res.pca <- PCA(Budgets, scale.unit = TRUE, graph = FALSE, ncp = 11)
```

On s'intéresse aux valeurs propres afin de voir la qualité de représentation (part d'inertie conservée) en fonction du nombre d'axes choisis.

```
get_eigenvalue(res.pca)
```

```
##          eigenvalue variance.percent cumulative.variance.percent
## Dim.1  4.972362e+00     4.520329e+01                     45.20329
## Dim.2  2.050638e+00     1.864217e+01                     63.84546
## Dim.3  1.290168e+00     1.172880e+01                     75.57426
## Dim.4  9.930553e-01     9.027775e+00                     84.60203
## Dim.5  7.083544e-01     6.439585e+00                     91.04162
## Dim.6  5.581499e-01     5.074090e+00                     96.11571
## Dim.7  2.042530e-01     1.856845e+00                     97.97255
## Dim.8  1.252021e-01     1.138201e+00                     99.11075
## Dim.9  6.281176e-02     5.710160e-01                     99.68177
## Dim.10 3.500069e-02     3.181881e-01                     99.99996
## Dim.11 4.502045e-06     4.092768e-05                    100.00000
```

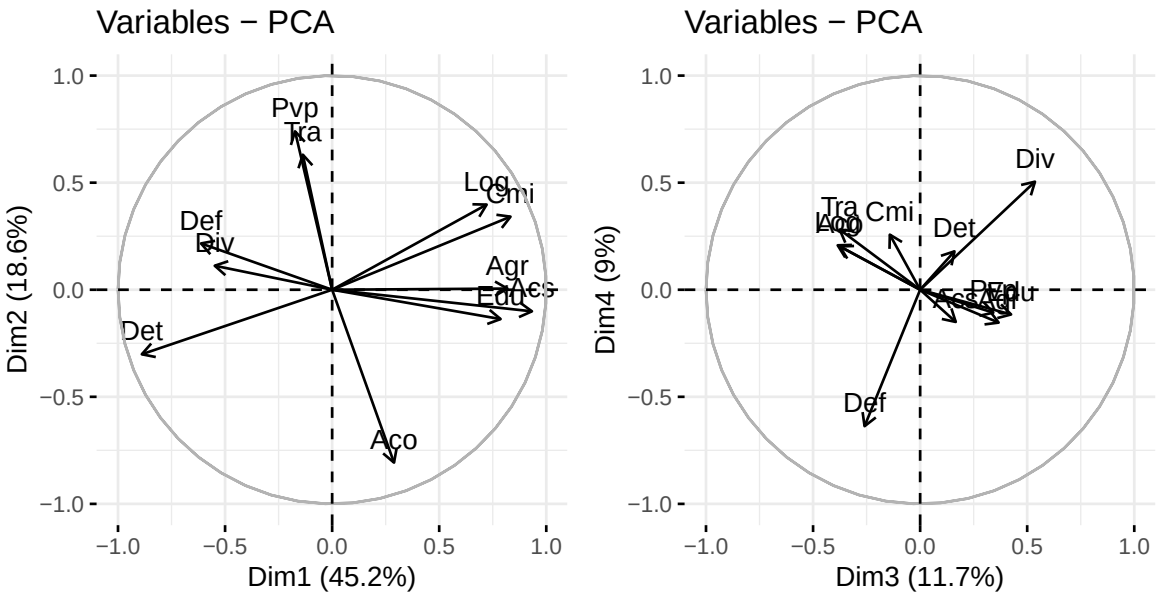
```
fviz_eig(res.pca, addlabels = TRUE, ylim = c(0,50))
```



On constate qu'on conserve environ 64% de l'inertie totale sur le premier plan principal et pour conserver presque 85%, il faut conserver 4 axes.

Nous nous intéressons à la représentation des variables.

```
p1=fviz_pca_var(res.pca, axes = 1:2)
p2=fviz_pca_var(res.pca, axes = 3:4)
grid.arrange(p1,p2,nrow=1)
```

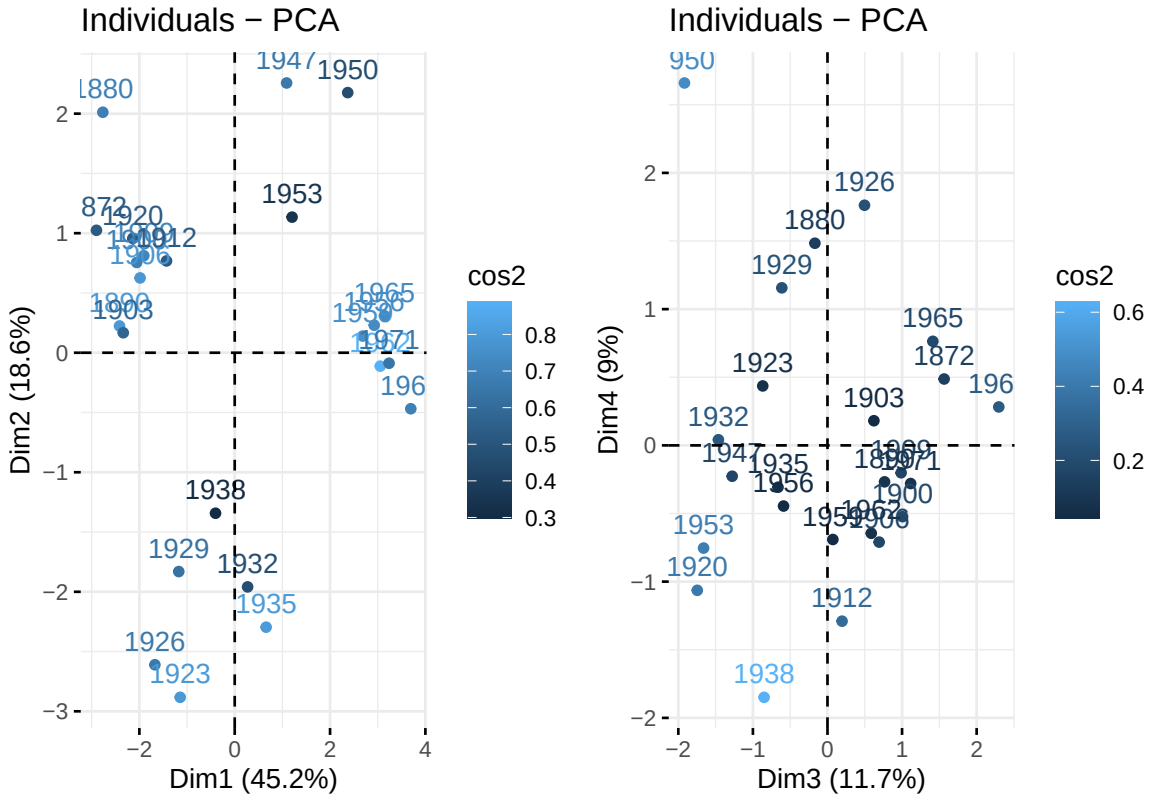


Nous constatons que sur le premier plan (composé des axes 1 et 2 appelés Dim. sur les graphiques, remarquons que les parts d'inertie conservée sur chaque axe sont rappelées en pourcentage) permet une bonne représentation de 9 des 11 variables initiales. Seules les variables `DEF` et `DIV` ne sont pas très bien représentées. On le voit également sur le plan

constitué des axes 3 et 4, on constate que ces deux variables ont des flèches plutôt longues ce qui signifie qu'elles sont corrélées aussi à ces axes. L'interprétation sur le premier plan est relativement aisée. Le premier axe est corrélé aux variables `LOG`, `CMI`, `AGR`, `ACS`, `EDU` et `DET` tandis que le second est corrélé aux variables `PVP`, `TRA` et `ANC`. Ces corrélations pouvant être positives et ou négatives. On comprendra alors qu'une année qui aura une coordonnée positive et grande sur le premier axe sera une année où l'état a dépensé une plus grande part de son budget qu'en moyenne pour les pôles de dépense `LOG`, `CMI`, `AGR`, `ACS`, `EDU` et moins pour `DET`. Et inversement si la coordonnée est négative. Ce graphique nous permet également de retrouver les corrélations, le groupe des variables `LOG`, `CMI`, `AGR`, `ACS`, `EDU` sont corrélées positivivement et ces variables sont corrélées négativement avec `DET`. Les variables `PVP` et `TRA` sont corrélées et toutes deux anticorrélées à `ANC`.

À présent, nous pouvons nous intéresser à la représentation des individus avec un code couleur donnant la qualité de représentation en fonction du $\cos^2$:

```
res.pca = res.pca <- PCA(Budgets, scale.unit = TRUE, graph = FALSE, ncp = 11)
p1=fviz_pca_ind(res.pca, axes = 1:2,col.ind="cos2")
p2=fviz_pca_ind(res.pca, axes = 3:4,col.ind="cos2")
grid.arrange(p1,p2,nrow=1)
```



La représentation sur le premier plan (axes 1 et 2) est évidemment la plus intéressante. Nous avons donné la représentation sur les axes 3 et 4 à titre d'exemple mais il n'apportera rien ici en terme d'interprétation. Sur ces données, un reprougrement assez net est observable entre les années proches chronologiquement. Il y a principalement 3 ou 4 groupes :

— un avec les années antérieures à la première guerre mondiale (à gauche) où on dépensait plus pour rembourser la dette de l'état, plus pour la défense et moins pour les politiques sociales d'éducation, de logement etc. ;
— un avec les années entre les 2 guerres mondiales, où l'on dépensait plus pour les anciens combattants et moins pour le travail et les pouvoirs publics (en bas);
— un avec les années des trentes glorieuses (à droite) où l'on dépensait moins pour rembourser la dette et plus dans des politiques agricoles, d'éducation de logement et de commerce.
— éventuellement un groupe avec les années 1947, 1950 et 1953 où l'on dépensait plus pour les travail et les pouvoirs publics.

Attention à l'interprétation, lorsque l'on écrit plus ou moins c'est par rapport à la moyenne des années.

Notons alors que l'ACP a permis d'avoir une représentation résumée efficace de ce jeu de données que les outils classiques (corrélations, nuages de points par rapport à un choix de 2 variables...) n'auraient pas permis.

Chapitre 2

Méthodes de classification non supervisée : K-means et CAH

2.1 Formulation du problème de classification non supervisée

Nous travaillons comme pour l'ACP sur un tableau de données tel que défini dans l'équation (1.1). Les variables sont supposées aussi être centrées et réduites.

Notre motivation est de former des groupes / classes / clusters d'individus se ressemblant sur la base des variables dont nous disposons. Cela pourrait permettre de repérer une organisation sous-jacente dans les données. La ressemblance entre individus sera fondée sur la distance dite euclidienne couramment employée en géométrie : pour deux individus U_i et $U_{i'}$ dans $\mathbb{R}^p$ on a la distance définie comme la norme de la différence entre les deux vecteurs :

$$d(U_i, U_{i'}) = \|U_i - U_{i'}\| = \sqrt{\sum_{j=1}^{p} (x_{ij} - x_{i'j})^2}\,.$$

Nous utilisons ici aussi la notion d'inertie :

$$I_T = \sum_{i=1}^{n} \|U_i - U_G\|^2$$

où $U_G = \frac{1}{n}\sum_{i=1}^{n} U_i$. Or si les données sont centrées, on a $U_G = 0$ (vecteur nul de $\mathbb{R}^p$). Par convention, l'inertie utilisée pour la classification supervisée ne comporte pas le terme $\frac{1}{n}$ utilisée dans l'inertie du nuage de point de l'ACP.

Nous cherchons une partition des U_i en K classes / groupes / clusters $(C_1, C_2, \ldots, C_K)$. Une partition vérifie que chaque U_i appartient à une et une unique classe. On notera

$$U_{C_k} = \frac{1}{|C_k|} \sum_{i \in C_k} U_i$$

le centre de gravité de la classe C_k où $|C_k|$ est le nombre d'individus dans C_k. On décompose l'inertie en commençant à sommer par classe :

$$
\begin{aligned}
I_T &= \sum_{i=1}^{n} \|U_i - U_G\|^2 = \sum_{k=1}^{K} \sum_{i \in C_k} \|U_i - U_G\|^2 \\
&= \sum_{k=1}^{K} \sum_{i \in C_k} \|(U_i - U_{C_k}) + (U_{C_k} - U_G)\|^2 \\
&= \sum_{k=1}^{K} \sum_{i \in C_k} \left(\|U_i - U_{C_k}\|^2 + \|U_{C_k} - U_G\|^2 + 2(U_i - U_{C_k}).(U_{C_k} - U_G) \right) \\
&= \sum_{k=1}^{K} \sum_{i \in C_k} \|U_i - U_{C_k}\|^2 + \sum_{k=1}^{K} |C_k| \|U_{C_k} - U_G\|^2 .
\end{aligned}
$$

Le passage de l'avant-dernière ligne à la dernière ligne se justifie par la bilinéarité du produit scalaire : $\sum_{i \in C_k} (U_i - U_{C_k}).(U_{C_k} - U_G) = (\sum_{i \in C_k} U_i - |C_k| \cdot U_{C_k}).(U_{C_k} - U_G) = 0.(U_{C_k} - U_G)$ et parce que $1/|C_k| \cdot \sum_{i \in C_k} U_i = U_{C_k}$.

Ainsi I_T se décompose en deux parties : $I_T = I_W + I_B$ avec

- $I_W = \sum_{k=1}^{K} \sum_{i \in C_k} \|U_i - U_{C_k}\|^2$ l'inertie intra-classes (W pour within) qui mesure l'hétérogénéité au sein des classes,
- $I_B = \sum_{k=1}^{K} |C_k| \|U_{C_k} - U_G\|^2$ l'inertie inter-classes (B pour between) qui mesure l'hétérogénéité entre les classes c'est-à-dire à quel point les classes sont bien séparées.

On s'accorde ainsi à définir une bonne partition comme une partition qui maximise l'inertie inter-classes I_B ou de manière équivalente qui minimise l'inertie intra-classes I_W. Formulé mathématiquement, on cherche une partition $\mathcal{C}^*$ pour K fixé telle que :

$$
\mathcal{C}^* = \arg \min_{(C_1, C_2, \ldots, C_K)} \sum_{k=1}^{K} \sum_{i \in C_k} \|U_i - U_{C_k}\|^2 .
$$

Il n'est pas possible d'explorer toutes les partitions possibles dont l'ordre de grandeur est K^n (pour chacun des n individus, on a K choix possibles de classe). On va avoir recours à deux algorithmes non exhaustifs : la méthodes des K-means et la classification ascendante hierarchique (CAH) qui viseront à obtenir une partition optimale.

2.2 Méthode des K-means

Cette méthode repose sur le choix de K centres ou moyennes (means en anglais). On affecte les individus aux classes associées aux centres dont ils sont le plus proche. On recalcule les centres comme le centre de gravité des individus et on itère jusqu'à convergence c'est-à-dire que les affectations des individus ne changent plus. Cet algorithme repose sur une initialisation aléatoire qui consiste à choisir les K centres.

Algorithme 1 (Algorithme des K-means)**.**

0. Initialisation, choix aléatoire de K centres : $(U_{C_1}, U_{C_2}, \ldots, U_{C_K})$.

On itère 1. et 2. jusqu'à convergence :

1. Affectation des individus aux K classes. $U_i \in C_k$ *si* $\|U_i - U_{C_k}\| = \min_l \|U_i - U_{C_l}\|$.

2. On recalcule les centres de classes : $U_{C_k} = \frac{1}{|C_k|} \sum_{i \in C_k} U_i$.

Proposition 3. *L'inertie intra-classes I_W diminue à chaque itération de l'algorithme. Ainsi il peut y avoir convergence vers un minimum local.*

D'après la proposition précédente, il est alors justifié de relancer plusieurs fois l'algorithme à partir d'initialisations différentes afin d'espérer avoir, parmi les partitions obtenues, la partition qui minimise globalement l'inertie intra-classes.

2.3 Classification ascendante hierarchique (CAH)

Le principe de la CAH est de commencer avec autant de classes qu'il y a d'individus ($K = n$). Chaque classe contient un unique individu. On fusionne ensuite les 2 classes les plus semblables (ici les 2 individus les plus proches) en une seule classe ce qui donne $K = n - 1$ classes. Et ainsi de suite jusqu'à n'obtenir qu'une seule classe ($K = 1$) contenant tous les individus. La classification est hierarchique car il y a emboîtement entre les partitions. En effet, la partition à K classes est obtenue comme la partition à $K + 1$ classes où 2 classes ont fusionné.

Pour pouvoir fusionner des classes, il faut une mesure de dissimiliraté entre classes. Nous utiliserons la distance de Ward (attention ce n'est pas une distance au sens strict mathématique) qui est cohérente avec la recherche d'une partition minimisant l'inertie intra-classes.

Définition 9 (Distance de Ward)**.** *Pour deux classes C_k et C_ℓ, on définit :*

$$d(C_k, C_\ell) = \frac{|C_k| \cdot |C_\ell|}{|C_k| + |C_\ell|} \|U_{C_k} - U_{C_\ell}\|^2 ,$$

qui correspond à l'augmentation d'inertie intra-classes (ou à la diminution d'inertie inter-classes) lorsqu'on fusionne les classes C_k et C_ℓ.

Nous montrons ci-dessous que la distance entre C_k et C_ℓ correspond à la perte d'inertie inter-classes lorsque l'on fusionne les classes C_k et C_ℓ pour passer de K à $K - 1$ classes. Nous notons $I_B^K - I_B^{K-1}$ cette perte d'inertie. Le centre de gravité de la classe fusionnée est la moyenne pondérée des centres de gravité de C_k et C_ℓ :

$$\frac{|C_k| \cdot U_{C_k} + |C_\ell| \cdot U_{C_\ell}}{|C_k| + |C_\ell|} .$$

On a alors :

$$\begin{aligned}
I_B^K - I_B^{K-1} &= |C_k| \cdot \|U_{C_k} - U_G\|^2 + |C_\ell| \cdot \|U_{C_\ell} - U_G\|^2 \\
&\quad -(|C_k| + |C_\ell|) \cdot \|\frac{|C_k| \cdot U_{C_k} + |C_\ell| \cdot U_{C_\ell}}{|C_k| + |C_\ell|} - U_G\|^2 \\
&= |C_k| \cdot \|U_{C_k} - U_G\|^2 + |C_\ell| \cdot \|U_{C_\ell} - U_G\|^2 \\
&\quad -\frac{1}{|C_k| + |C_\ell|}\||C_k| \cdot (U_{C_k} - U_G) + |C_\ell| \cdot (U_{C_\ell} - U_G)\|^2 \\
&= |C_k| \cdot \|U_{C_k} - U_G\|^2 + |C_\ell| \cdot \|U_{C_\ell} - U_G\|^2 \\
&\quad -\frac{1}{|C_k| + |C_\ell|}\left(|C_k|^2\|U_{C_k} - U_G\|^2 + |C_\ell|^2\|U_{C_\ell} - U_G\|^2 + 2|C_k||C_\ell|(U_{C_k} - U_G).(U_{C_\ell} - U_G)\right) \\
&= \frac{|C_k||C_\ell|}{|C_k| + |C_\ell|}\left(\|U_{C_k} - U_G\|^2 + \|U_{C_\ell} - U_G\|^2 - 2(U_{C_k} - U_G).(U_{C_\ell} - U_G)\right) \\
&= \frac{|C_k||C_\ell|}{|C_k| + |C_\ell|}\|(U_{C_k} - U_G) - (U_{C_\ell} - U_G)\|^2 = \frac{|C_k||C_\ell|}{|C_k| + |C_\ell|}\|U_{C_k} - U_{C_\ell}\|^2 .
\end{aligned}$$

L'algorithme peut alors s'écrire ainsi :

Algorithme 2.
Initialisation : former $K = n$ classes avec un unique individu par classe.

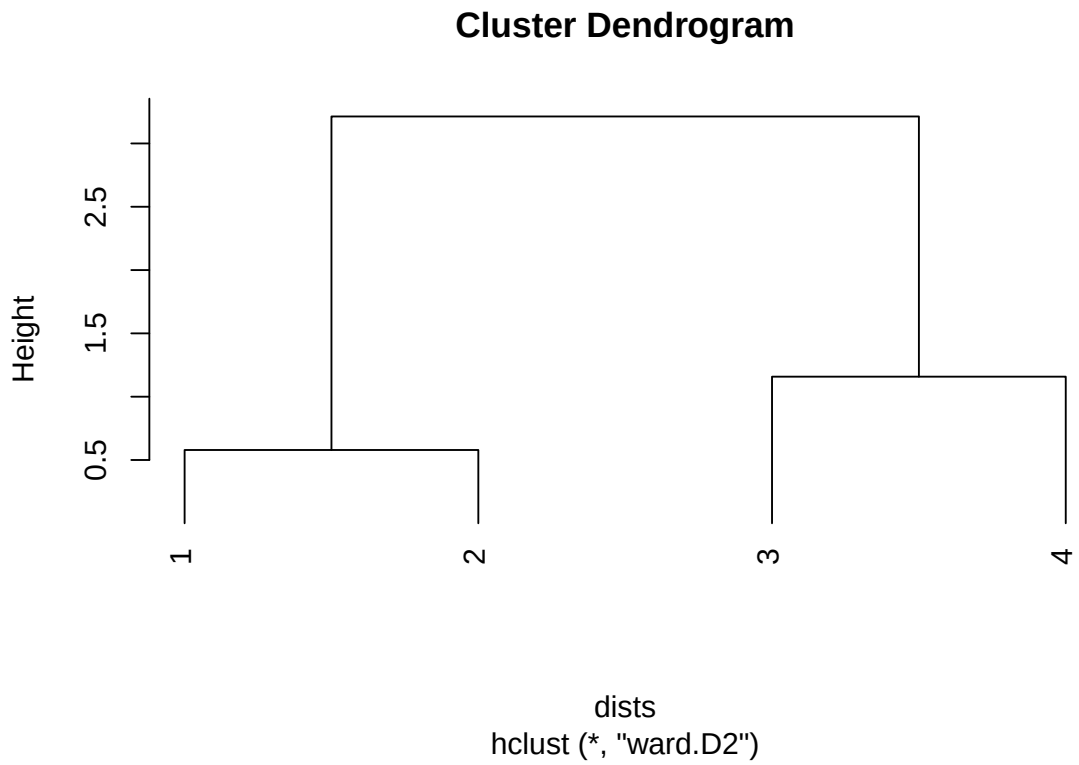
Pour $K = n$ à $K = 2$, faire :

- *Calculer les $K(K-1)/2$ distances de Ward entre toutes les paires de classes ;*

- *Fusionner les 2 classes les plus proches (au sens de la distance de Ward), ainsi $K < -K - 1$.*

Au fur et à mesure des fusions, l'inertie intra-classes augmente et l'inertie inter-classes diminue. On part d'une inertie intra-classes $I_W = 0$ lorsque $K = n$, les classes étant homogènes puisque ne contenant qu'un seul individu, et on arrive à $I_W = I_T$ lorsque l'on a une unique classe $K = 1$.

Afin de représenter ces fusions successives, on peut afficher un dendrogramme c'est-à-dire un arbre dont les feuilles sont les individus et dont les branches matérialisent les regroupements.



On présente ci-dessus un exemple de dendrogramme sur $n = 4$ points. On constate que les individus 1 et 2 ont d'abord été regroupés une une classe puis les individus 3 et 4 et enfin les 2 classes contenant chacune 2 individus ont été fusionnées. La hauteur à laquelle les branches se rejoignent est liée (par une formule obscure, voir dans l'exemple ci-dessous) à la perte d'inertie due à la fusion. Ainsi un saut dans la hauteur des fusions peut indiquer que l'on essaie de regrouper des classes trop différentes.

Remarque 2. *Notons que la CAH peut être employée avec d'autres choix de distance entre individus que la distance euclidienne et d'autres choix de dissimilirité entre classes que la distance de Ward.*

2.4 Application aux données du budget de l'état

Nous reprenons le tableau de données des budgets de l'état. Il faut bien penser à centrer-réduire les données avant d'appliquer l'algorithme des K-means ou la CAH (dans l'ACP, le centrage et la réduction sont faits par défaut).

```
BudgetsCR = scale(Budgets)
```

Nous commençons par appliquer l'algorithme des $K - means$, nous choisissons $K = 4$ classes. Nous fixons `centers=4` pour ce faire et `nstart=200` pour lancer 200 initialisations.

```
kmBudgets = kmeans(BudgetsCR,centers=4,nstart = 200)
```

Nous avons accès à l'inertie totale, inter-classes et intra-classes grâce aux commandes (`ss`

signifie sum of squares ce qui correspond bien aux termes d'inertie):

```
kmBudgets$totss
```

```
## [1] 253
```

```
kmBudgets$betweenss
```

```
## [1] 169.0026
```

```
kmBudgets$tot.withinss
```

```
## [1] 83.99736
```

Afin d'obtenir les classes, on peut utiliser la commande :

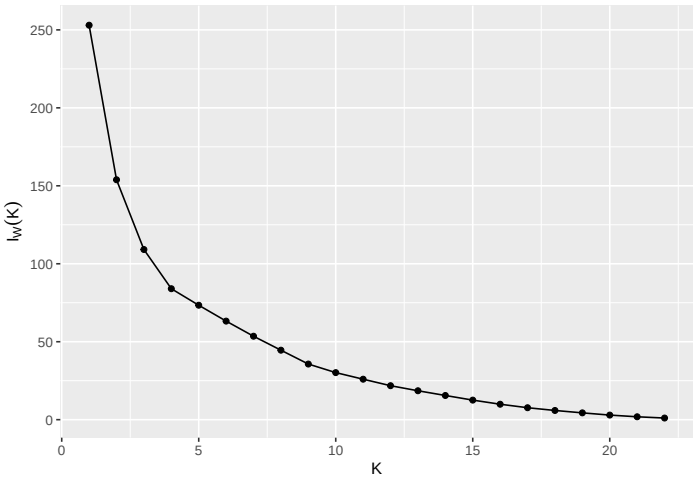
```
kmCluster = kmBudgets$cluster
kmCluster
```

```
## 1872 1880 1890 1900 1903 1906 1909 1912 1920 1923 1926 1929 1932 1935 1938 1947
##    3    3    3    3    3    3    3    3    3    1    1    1    1    1    1    2
## 1950 1953 1956 1959 1962 1965 1968 1971
##    2    2    4    4    4    4    4    4
```

On obtient un vecteur de taille $n = 24$ qui donne un numéro de classe pour chaque individu. 2 individus avec le même numéro de classe sont dans la même classe.

Afin de choisir le nombre de classes K, on peut représenter l'inertie intra-classes en fonction du nombre de classes

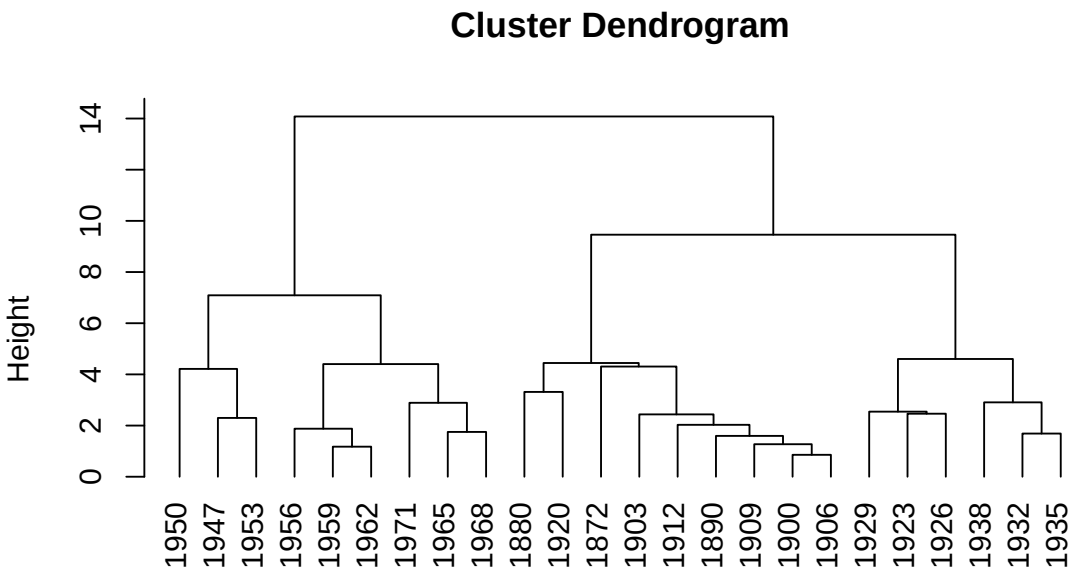
```
Kmax = 22
IW=numeric(Kmax)
for (k in 1:(Kmax))
{
  km = kmeans(BudgetsCR,centers=k,nstart = 100)
  IW[k] = km$tot.withinss
}
ggplot(data.frame(k = 1:(Kmax), # On créée un tableau avec nos deux colonnes
                  Iw = IW)) +
  aes(x = k, y = Iw) + # On spécifie les axes de représentation
  geom_point() + # On trace les points
  geom_line() + # On les relie par une ligne
  labs(x = "K", # On nomme les axes
       y = expression(I[W](K)))
```



On peut utiliser un critère du "coude" pour choisir K. Lorsque l'on augmente le nombre de classes, on observe une décroissante forte de l'inertie intra-classes au début (pour un petit K) puis cette décroissante s'atténue au niveau de ce qu'on appelle un coude, là où l'on observe un changement de pente. Sur ces données ce coude se situe autour de $K = 4$ ce qui justifie ce choix.

Pour appliquer, la CAH on doit calculer les distances entre individus. On affiche ensuite le dendrogramme.

```
BudDist = dist(BudgetsCR)
CAHBud = hclust(BudDist,method="ward.D2")
plot(CAHBud,hang=-1)
```



D'après le dendrogramme, on peut également justifier le choix de $K = 4$ car le saut pour passer de $K = 4$ à $K = 3$ est plus important que les sauts précédents. On peut ainsi retrouver les inerties intra-classes pour les différents K grâce à la formule ci-dessous reliant cette inertie aux hauteurs des branches du dendrogramme :

```
distancesWard = 0.5 * CAHBud$height^2 # Hauteurs des branches (29)
IWCAH = rev(c(0, cumsum(distancesWard)))
IWCAH
```

```
##  [1] 253.0000000 153.8776549 109.1352472  83.9973560  73.4091903  63.5289377
##  [7]  53.8361128  44.5645707  35.6835857  30.1905247  25.9651023  21.7903053
## [13]  18.5550484  15.5210633  12.5527083   9.9127609   7.8548932   6.0929687
## [19]   4.5602308   3.1375633   1.8640991   1.0565767   0.3670159   0.0000000
```

Le vecteur `IWCAH` contient pour $K = 1$ à $K = 24$ les inerties intra-classes. On constate que pour $K = 4$, l'inertie intra-classes est 83.997 qui correspond à l'inertie intra-classes obtenue pour le même nombre de classes avec les K-means. On a sans doute obtenue les 2 mêmes partitions avec les 2 algorithmes mais ce n'est pas toujours le cas si les données sont moins bien séparées.

On extrait les clusters pour la CAH et on compare avec ceux obtenus pour les K-means par un tableau d'effectif :

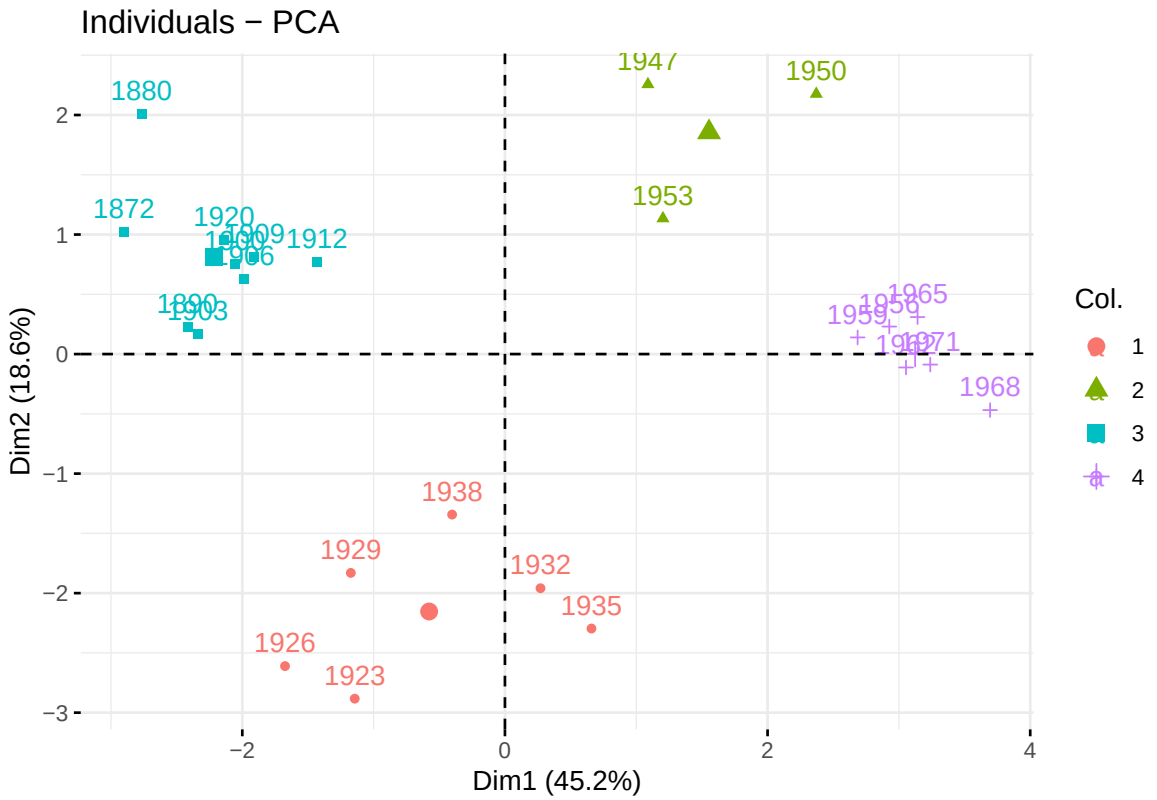
```
CAHCluster = cutree(CAHBud,k=4)
kable(table(CAHCluster,kmCluster))
```

1	2	3	4
0	0	9	0
6	0	0	0
0	3	0	0
0	0	0	6

On voit qu'à renumérotation près des classes, il y a une correspondance parfaite entre les 2 partitions.

On peut la représenter par une couleur spécifique sur le premier plan de l'ACP :

```
res.pca <- PCA(Budgets, scale.unit = TRUE, graph = FALSE, ncp = 11)
fviz_pca_ind(res.pca, axes = 1:2,col.ind=factor(kmCluster))
```



On retrouve les groupes d'années qu'on avait identifiés précédemment. Rappelons que le problème est ici relativement simple et les données en nombre raisonnable. Sur un jeu de données plus massif, le recours à ces algorithmes est indispensable.

Deuxième partie

Apprentissage supervisé

Chapitre 3

Modèle linéaire

3.1 Introduction

Nous introduisons dans ce chapitre un modèle très utile lorque l'on veut analyser, modéliser des données ou faire de la prédiction à partir de ces données : il s'agit du modèle linéaire. Il permet de modéliser une variable **quantitative** *i.e.* une variable qui varie continûment dans un intervalle donné (ex : la taille, le poids, ...) en fonction d'autres variables dites **explicatives** qui peuveut être quantitatives ou qualitatives *i.e.* ne prendre qu'un nombre fini fixe de valeurs (ex : le genre d'une personne qui peut être masculin ou féminin, le type d'un traitement dont on veut comprendre l'effet sur le rendement de certaines cultures, ...).

Dans la suite du cours nous étudierons différents types de modèles linéaires :

- la régression linéaire simple lorsque l'on souhaite expliquer une variable quantitative à l'aide d'une seule variable explicative quantitative (rappels de la 1ère année) ;
- la régression linéaire multiple lorsque l'on souhaite expliquer une variable quantitative à l'aide de plusieurs variables explicatives quantitatives (dans ce chapitre puis à la Section 4.1 du Chapitre 4) ;
- l'ANOVA (ANalysis Of VAriance) à 1 facteur lorsque l'on souhaite expliquer une variable quantitative à l'aide d'une variable explicative qualitative (dans ce chapitre puis à la Section 4.2 du Chapitre 4) ;
- l'ANOVA à 2 facteurs lorsque l'on souhaite expliquer une variable quantitative à l'aide de deux variables explicatives qualitatives (voir Section 4.3 du Chapitre 4) ;
- l'ANCOVA lorsque l'on souhaite expliquer une variable quantitative à l'aide de variables explicatives quantitatives et qualitatives (voir Section 4.4 du Chapitre 4).

3.2 Modèle

3.2.1 Rappels : régression linéaire simple (vue en 1ère année)

3.2.1.1 Exemple : relation entre masse du cerveau et volume de la tête

On dispose de la masse du cerveau en grammes et du volume de la tête en cm^3 de $n = 237$ individus adultes et on cherche à expliquer la masse du cerveau : `Masse` (variable quantitative

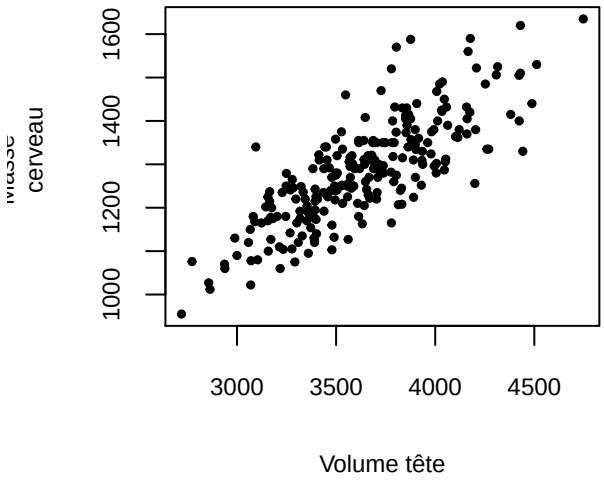
à expliquer : Y) en fonction du volume de la tête : `Volume` (variable explicative quantitative : x). Voici un extrait des données :

```
brainhead <- read.table("brainhead.dat", quote="\"")
colnames(brainhead)=c('','','Volume','Masse')
head(brainhead[,c(3,4)])
```

```
##   Volume Masse
## 1   4512  1530
## 2   3738  1297
## 3   4261  1335
## 4   3777  1282
## 5   4177  1590
## 6   3585  1300
```

Les données sont représentées dans le graphe ci-dessous.

```
plot(brainhead$Volume,brainhead$Masse,pch=20,xlab="Volume tête",ylab="Masse
     cerveau", cex.lab=0.75, cex.axis=0.75,cex=0.75)
```



On peut donc utiliser un modèle de régression linéaire simple pour modéliser ces données.

3.2.1.2 Écriture du modèle de régression linéaire simple

Le modèle de régression linéaire simple s'écrit comme suit dans le cas de la relation entre la masse du cerveau en grammes et du volume de la tête :

$$Y_i = a + bx_i + E_i,\ 1 \leq i \leq n. \tag{3.1}$$

où

- Y_i est une variable aléatoire modélisant la masse du cerveau du i-ème individu,
- les E_i sont des variables aléatoires (v.a.) indépendantes et identiquement distribuées (i.i.d.) de loi $\mathcal{N}(0, \sigma^2)$ qui modélisent les erreurs de mesure et/ou de modélisation,
- x_i est le volume de la tête du i-ème individu,
- a et b sont des paramètres réels inconnus qu'il va falloir estimer.

Une autre façon d'écrire (3.1) est la suivante :

$$\text{les } Y_i \text{ sont indépendantes de loi } \mathcal{N}(a + bx_i, \sigma^2).$$

Le modèle (3.1) peut aussi se réécrire sous la forme matricielle suivante :

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\theta} + \mathbf{E}, \tag{3.2}$$

où

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, \quad \boldsymbol{\theta} = \begin{pmatrix} a \\ b \end{pmatrix}, \quad \mathbf{E} = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{pmatrix}.$$

Cette écriture matricielle est en fait générique et tous les modèles linéaires s'écrivent de cette façon avec des matrices $\mathbf{X}$ et des vecteurs de paramètres $\boldsymbol{\theta}$ qui changent en fonction des modèles.

3.2.2 Écriture du modèle linéaire dans le cas général

De façon générale, un modèle linéaire s'écrit de la façon suivante :

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\theta} + \mathbf{E}, \tag{3.3}$$

où

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,p} \\ x_{2,1} & \cdots & x_{2,p} \\ \vdots & \cdots & \vdots \\ x_{n,2} & \cdots & x_{n,p} \end{pmatrix}, \quad \boldsymbol{\theta} = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_p \end{pmatrix}, \quad \mathbf{E} = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{pmatrix}.$$

Avec ces notations, $\mathbf{Y}$ et $\mathbf{E}$ sont des vecteurs colonnes de taille $n \times 1$, $\mathbf{X}$ est une matrice ayant n lignes et p colonnes et $\boldsymbol{\theta}$ est un vecteur colonne de taille $p \times 1$.

Comme précédemment, dans (3.3) :

- les E_i sont des variables aléatoires (v.a.) indépendantes identiquement distribuées (i.i.d.) gaussiennes d'espérance nulle et de variance σ^2 (les E_i sont i.i.d. de loi $\mathcal{N}(0, \sigma^2)$),
- les θ_k sont des paramètres inconnus à estimer.

De façon équivalente, on peut écrire :

$$Y_i = \theta_1 x_{i,1} + \theta_2 x_{i,2} + \cdots + \theta_p x_{i,p} + E_i, \ \forall 1 \leq i \leq n.$$

On remarque ainsi que pour tout i, Y_i s'écrit comme une combinaison linéaire des $x_{i,k}$ qui correspond à la valeur de la k-ième variable explicative pour la i-ème observation ou l'individu i d'où le nom de modèle linéaire.

Avec les hypothèses précédentes sur les E_i, on a que :

$$\text{les } Y_i \text{ sont des v.a. indépendantes de loi } \mathcal{N}(\theta_1 x_{i,1} + \theta_2 x_{i,2} + \cdots + \theta_p x_{i,p}, \sigma^2). \tag{3.4}$$

Ainsi chaque observation de la variable à expliquer est modélisée comme la réalisation d'une variable aléatoire gaussienne dont l'espérance est une combinaison linéaire des variables explicatives et de variance σ^2.

3.2.2.1 Exemple 1 : modèle de régression linéaire multiple

Il s'agit d'une généralisation de la régression linéaire simple : au lieu d'avoir une seule variable explicative on en a plusieurs.

Exemple : chenille processionnaire du pin. Cette chenille se développe de préférence sur des pins et peut causer des dégâts considérables. On souhaite donc étudier l'influence de certaines caractéristiques de peuplements forestiers sur le développement de la chenille processionnaire.

FIGURE 3.1 – Photographie d'une chenille processionnaire du pin.

On dispose de $n = 33$ parcelles d'une surface de 10 hectares. Chaque parcelle est alors échantillonnée en placettes de 5 ares. Sur chaque placette on mesure :

- l'altitude
- la pente
- le nombre de pins dans une placette
- la hauteur de l'arbre au centre de la placette
- le diamètre de cet arbre
- ....

Les valeurs données pour une parcelle sont les moyennes des mesures sur l'ensemble des placettes de la parcelle. Ici il y a 10 variables explicatives et la variable à expliquer y est la variable `NbNids` correspondant au nombre de nids.

```
chenilles <- read.table("chenilles.txt", header=TRUE, sep=" ")
dim(chenilles)
```

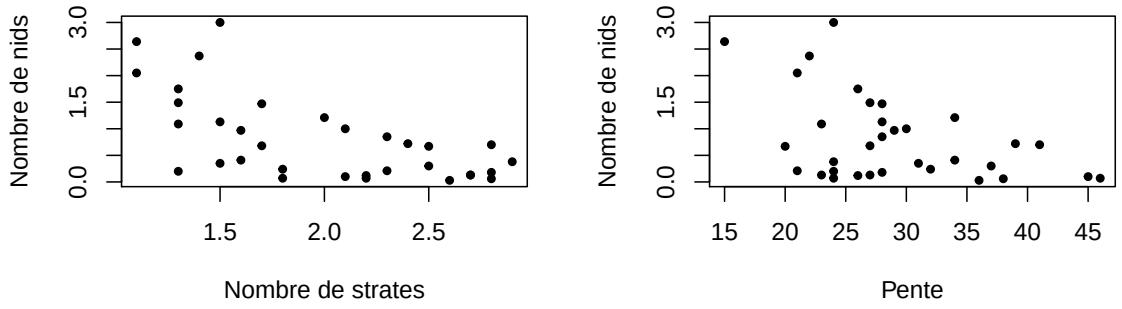
```
## [1] 33 12
```

```
head(chenilles)
```

```
##   Altitude Pente NbPins Hauteur Diametre Densite Orient HautMax NbStrat Melange
## 1     1200    22      1     4.0     14.8     1.0    1.1     5.9     1.4     1.4
## 2     1342    28      8     4.4     18.0     1.5    1.5     6.4     1.7     1.7
## 3     1231    28      5     2.4      7.8     1.3    1.6     4.3     1.5     1.7
## 4     1254    28     18     3.0      9.2     2.3    1.7     6.9     2.3     1.6
## 5     1357    32      7     3.7     10.7     1.4    1.7     6.6     1.8     1.3
## 6     1250    27      1     4.4     14.8     1.0    1.7     5.8     1.3     1.4
##   NbNids  LogNids
## 1   2.37  0.86289
## 2   1.47  0.38526
## 3   1.13  0.12222
## 4   0.85 -0.16252
## 5   0.24 -1.42712
## 6   1.49  0.39878
```

On représente le nombre de nids en fonction du nombre de strates et en fonction de la pente par exemple.

```
par(mfrow=c(1,2))
plot(chenilles$NbStrat,chenilles$NbNids,xlab='Nombre de strates',
     ylab='Nombre de nids',pch=20,type='p')
plot(chenilles$Pente,chenilles$NbNids,xlab='Pente',
     ylab='Nombre de nids',pch=20,type='p')
```



Dans ce cas, le modèle de régression linéaire multiple s'écrit :

$$Y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_p x_{i,p} + E_i,\ 1 \leq i \leq n, \tag{3.5}$$

où :

- Y_i est la variable aléatoire modélisant le nombre de nids dans la parcelle i,
- $x_{i,k}$ est la valeur de la k-ième variable pour la parcelle i,
- les E_i sont i.i.d. de loi $\mathcal{N}(0, \sigma^2)$.

On retrouve la forme générale (3.3) avec l'écriture matricielle suivante :

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\theta} + \mathbf{E}, \tag{3.6}$$

où

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_{1,1} & \cdots & x_{1,p} \\ 1 & x_{2,1} & \cdots & x_{2,p} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_{n,2} & \cdots & x_{n,p} \end{pmatrix}, \quad \boldsymbol{\theta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \mathbf{E} = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{pmatrix}.$$

3.2.2.2 Exemple 2 : modèle d'ANOVA à 1 facteur

Exemple : influence du genre d'un individu sur sa taille

On souhaite vérifier statistiquement si le genre d'un individu a une influence sur sa taille. La variable quantitative à expliquer est donc la taille et la variable explicative est le genre. Il s'agit d'une variable qualitative ne prenant que deux valeurs (ou deux modalités) : "femme" et "homme". Dans cette étude, il y a 29 mesures faites sur des femmes et 38 sur des hommes.

```
PT <- read.table("PT.txt", header=TRUE, sep="\t", dec=",")
head(PT)
```

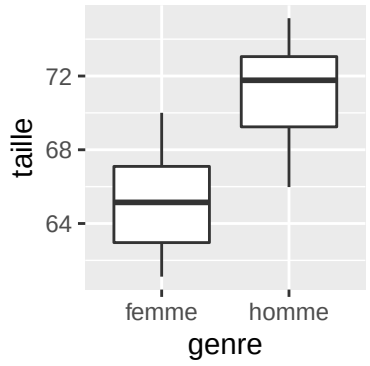
```
##   genre   taille    poids
## 1 femme 61.11732 138.5426
## 2 femme 62.10180 127.9359
## 3 femme 62.11668 119.4516
## 4 femme 62.13176 109.8725
## 5 femme 61.80914 107.1128
## 6 femme 62.09234 104.1216
```

```
table(PT$genre)
```

```
##
## femme homme
##    29    38
```

On peut représenter les données en utilisant un boxplot.

```
library(ggplot2)
ggplot(PT,aes(x=genre,y=taille))+geom_boxplot()
```



Pour valider statistiquement le fait que le genre a effectivement un effet sur la taille nous

allons utiliser le modèle d'ANOVA à 1 facteur. Pour cela, nous allons modéliser les données comme suit.

On note $Y_{i,k}$ la variable aléatoire qui modélise la taille du k-ième individu ayant le genre i. Ici, $1 \leq i \leq I = 2$ et $1 \leq k \leq n_i$, où n_i représente le nombre d'individus ayant le genre i. Si on note "1" le genre "femme" et "2" le genre "homme" : $n_1 = 29$ et $n_2 = 38$.

Le modèle d'ANOVA à 1 facteur s'écrit :

$$Y_{i,k} = \mu + \alpha_i + E_{i,k}, \quad 1 \leq i \leq I,\ 1 \leq k \leq n_i, \tag{3.7}$$

où les $E_{i,k}$ sont i.i.d.,de loi $\mathcal{N}(0, \sigma^2)$, μ décrit l'effet moyen et α_i représente l'effet additif dû à la modalité i du facteur. Dans ce modèle, il y a $I + 1$ paramètres : μ, α_1, $\alpha_2, \ldots, \alpha_I$.

Ce modèle s'écrit bien sous la forme matricielle (3.3) avec

$$\mathbf{Y} = \begin{pmatrix} Y_{1,1} \\ \vdots \\ Y_{1,n_1} \\ Y_{2,1} \\ \vdots \\ Y_{2,n_2} \\ \vdots \\ Y_{I,1} \\ \vdots \\ Y_{I,n_I} \end{pmatrix}, \ \mathbf{X} = \begin{pmatrix} 1 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ \vdots & 1 & 0 & \ldots & 0 \\ \vdots & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ \vdots & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & 0 & 0 & \ldots & 1 \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 1 & 0 & 0 & \ldots & 1 \end{pmatrix}, \ \boldsymbol{\theta} = \begin{pmatrix} \mu \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_I \end{pmatrix}, \ \mathbf{E} = \begin{pmatrix} E_{1,1} \\ \vdots \\ E_{1,n_1} \\ E_{2,1} \\ \vdots \\ E_{2,n_2} \\ \vdots \\ E_{I,1} \\ \vdots \\ E_{I,n_I} \end{pmatrix},$$

où $\mathbf{Y}$ et $\mathbf{E}$ sont des vecteurs colonnes ayant n lignes, $\boldsymbol{\theta}$ est de taille $(I+1) \times 1$ (vecteur colonne ayant $I + 1$ lignes) et $\mathbf{X}$ est une matrice de taille $n \times (I + 1)$ n'ayant que des 1 dans la 1ère colonne et telle que la valeur 1 est présente n_i fois dans la $(i + 1)$-ième colonne de la matrice.

3.3 Estimation des paramètres

3.3.1 Estimation des θ_k dans (3.3)

Les paramètres θ_k sont estimés en utilisant la méthode du maximum de vraisemblance. En utilisant l'hypothèse (3.4), cette méthode consiste à maximiser par rapport aux θ_k et σ^2 le critère suivant :

$$\begin{aligned} L(Y_1, \ldots, Y_n; \theta_1, \ldots, \theta_p, \sigma^2) &= \prod_{i=1}^{n} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2} \left(Y_i - \theta_1 x_{i,1} - \cdots - \theta_p x_{i,p} \right)^2 \right] \right\} \\ &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(Y_i - \theta_1 x_{i,1} - \cdots - \theta_p x_{i,p} \right)^2 \right]. \end{aligned}$$

En passant au log, on obtient que maximiser la log-vraisemblance par rapport aux θ_k à σ^2 fixé revient à minimiser par rapport aux θ_k le critère des moindres carrés :

$$(\widehat{\theta}_1, \ldots, \widehat{\theta}_p) = \text{Argmin}_{(\theta_1,\ldots,\theta_p)\in\mathbb{R}^p} \sum_{i=1}^{n} (Y_i - \theta_1 x_{i,1} - \cdots - \theta_p x_{i,p})^2 .$$

Ce critère peut se réécrire comme suit :

$$\widehat{\boldsymbol{\theta}} = \text{Argmin}_{\boldsymbol{\theta}\in\mathbb{R}^p} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\theta}\|^2 = \text{Argmin}_{\theta\in\mathbb{R}^p} (\mathbf{Y} - \mathbf{X}\boldsymbol{\theta})' (\mathbf{Y} - \mathbf{X}\boldsymbol{\theta}) , \tag{3.8}$$

où A' désigne la transposée de A, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)'$, $\widehat{\boldsymbol{\theta}} = (\widehat{\theta}_1, \ldots, \widehat{\theta}_p)'$ et $\|u\|^2 = \sum_{i=1}^n u_i^2$ désigne le carré de la norme euclidienne standard pour $u \in \mathbb{R}^n$.

Avant d'établir la proposition 4 sur l'estimateur du maximum de vraisemblance de $\boldsymbol{\theta}$ défini par (3.8), nous donnons des définitions et propriétés sur les vecteurs aléatoires.

Définition 10 (Vecteur aléatoire)**.** *On dit que $\boldsymbol{U} = (U_1, \ldots, U_p)'$ est un vecteur aléatoire lorsque chaque composante U_i est une variable aléatoire.*

Définition 11 (Espérance d'un vecteur aléatoire)**.** *Soit $\boldsymbol{U} = (U_1, \ldots, U_p)'$ un vecteur aléatoire. On définit son espérance par : $\mathbb{E}(\boldsymbol{U}) = (\mathbb{E}(U_1), \ldots, \mathbb{E}(U_p))'$*

Définition 12 (Matrice de covariance d'un vecteur aléatoire)**.** *Soit $\boldsymbol{U} = (U_1, \ldots, U_p)'$ un vecteur aléatoire. On définit sa matrice de covariance de U par :*

$$\text{Cov}(\boldsymbol{U}) = \mathbb{E}[(\boldsymbol{U} - \mathbb{E}(\boldsymbol{U}))(\boldsymbol{U} - \mathbb{E}(\boldsymbol{U}))'].$$

Autrement dit,

$$\text{Cov}(\boldsymbol{U}) = (\text{Cov}(U_i, U_j))_{1\le i,j\le p}.$$

Propriétés 1. *Soit A une matrice de taille $m \times p$, on a que :*

$$\mathbb{E}(\boldsymbol{A}\,\boldsymbol{U}) = \boldsymbol{A}\mathbb{E}(\boldsymbol{U}) \text{ et } \text{Cov}(\boldsymbol{A}\,\boldsymbol{U}) = \boldsymbol{A}\,\text{Cov}(\boldsymbol{U})\boldsymbol{A}'.$$

Ces propriétés viennent de la définition du produit d'une matrice par un vecteur et de la linéarité de l'espérance.

Proposition 4. *L'estimateur de $\boldsymbol{\theta}$ dans le modèle* (3.3) *défini par (3.8) satisfait :*

$$(\boldsymbol{X}'\boldsymbol{X})\widehat{\boldsymbol{\theta}} = \boldsymbol{X}'\boldsymbol{Y}, \tag{3.9}$$

et sous l'hypothèse que $\boldsymbol{X}'\boldsymbol{X}$ est inversible

$$\widehat{\boldsymbol{\theta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y} .$$

Dans ce cas, $\widehat{\boldsymbol{\theta}}$ ainsi défini est un estimateur sans biais. De plus, la matrice de covariance de $\widehat{\boldsymbol{\theta}}$ définie par $\text{Cov}(\widehat{\boldsymbol{\theta}}) = \mathbb{E}[(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta})']$ est telle que :

$$\text{Cov}(\widehat{\boldsymbol{\theta}}) = \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1}.$$

Preuve de la proposition 4

Soit

$$F(\theta_1,\ldots,\theta_p)=\sum_{i=1}^{n}\left(Y_i-\theta_1x_{i,1}-\cdots-\theta_px_{i,p}\right)^2.$$

On a alors :

$$\frac{\partial F}{\partial\theta_1}=-2\sum_{i=1}^{n}x_{i,1}\left(Y_i-\theta_1x_{i,1}-\cdots-\theta_px_{i,p}\right)=-2\left\langle\mathbf{Y}-\mathbf{X}\boldsymbol{\theta};\begin{pmatrix}x_{1,1}\\x_{2,1}\\\vdots\\x_{n,1}\end{pmatrix}\right\rangle=-2\begin{pmatrix}x_{1,1}\\x_{2,1}\\\vdots\\x_{n,1}\end{pmatrix}'(\mathbf{Y}-\mathbf{X}\boldsymbol{\theta})$$

$$\vdots$$

$$\frac{\partial F}{\partial\theta_p}=-2\sum_{i=1}^{n}x_{i,p}\left(Y_i-\theta_1x_{i,1}-\cdots-\theta_px_{i,p}\right)=-2\left\langle\mathbf{Y}-\mathbf{X}\boldsymbol{\theta};\begin{pmatrix}x_{1,p}\\x_{2,p}\\\vdots\\x_{n,p}\end{pmatrix}\right\rangle=-2\begin{pmatrix}x_{1,p}\\x_{2,p}\\\vdots\\x_{n,p}\end{pmatrix}'(\mathbf{Y}-\mathbf{X}\boldsymbol{\theta}),$$

où $<\cdot,\cdot>$ désigne le produit scalaire usuel dans $\mathbb{R}^n$. Puisque $\widehat{\boldsymbol{\theta}}=(\widehat{\theta}_1,\ldots,\widehat{\theta}_p)'$ satisfait :

$$\frac{\partial F}{\partial\theta_k}(\widehat{\theta}_1,\ldots,\widehat{\theta}_p)=0,\ \forall k\in\{1,\ldots,p\},$$

on déduit des calculs précédents que $\boldsymbol{\theta}$ vérifie :

$$\mathbf{X}'(\mathbf{Y}-\mathbf{X}\widehat{\boldsymbol{\theta}})=0\ \textit{i.e.}\ \mathbf{X}'\mathbf{X}\widehat{\boldsymbol{\theta}}=\mathbf{X}'\mathbf{Y}.$$

Donc, lorsque $\mathbf{X}'\mathbf{X}$ est inversible, on a bien :

$$\widehat{\boldsymbol{\theta}}=(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}.$$

On a donc :

$$\widehat{\boldsymbol{\theta}}=(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{X}\boldsymbol{\theta}+\mathbf{E})=\boldsymbol{\theta}+(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{E}$$

d'où l'on déduit que $\widehat{\boldsymbol{\theta}}$ est sans biais. De plus,

$$\mathrm{Cov}(\widehat{\boldsymbol{\theta}})=\mathbb{E}[(\widehat{\boldsymbol{\theta}}-\boldsymbol{\theta})(\widehat{\boldsymbol{\theta}}-\boldsymbol{\theta})']=\mathbb{E}[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{E}\mathbf{E}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}]=\sigma^2(\mathbf{X}'\mathbf{X})^{-1},$$

car $\mathbb{E}(\mathbf{E}\mathbf{E}')=\sigma^2\mathrm{Id}_{\mathbb{R}^n}$ où $\mathrm{Id}_{\mathbb{R}^n}$ désigne la matrice identité dans $\mathbb{R}^n$.

Remarque 3. *On peut obtenir* (3.9) *en utilisant une preuve plus "geométrique".*

Par définition de $\widehat{\boldsymbol{\theta}}$ *donnée par* (3.8)*,* $\boldsymbol{X}\widehat{\boldsymbol{\theta}}$ *correspond à la projection orthogonale de* $\boldsymbol{Y}$ *sur l'image de* $\boldsymbol{X}$ *correspondant au sous-espace vectoriel engendré par les colonnes de* $\boldsymbol{X}$*.* $\boldsymbol{X}\widehat{\boldsymbol{\theta}}$ *satisfait donc les équations suivantes :*

$$\begin{aligned}\langle\boldsymbol{Y}-\boldsymbol{X}\widehat{\boldsymbol{\theta}},\boldsymbol{X}_1\rangle&=0,\\\langle\boldsymbol{Y}-\boldsymbol{X}\widehat{\boldsymbol{\theta}},\boldsymbol{X}_2\rangle&=0,\\&\vdots\\\langle\boldsymbol{Y}-\boldsymbol{X}\widehat{\boldsymbol{\theta}},\boldsymbol{X}_{p+1}\rangle&=0,\end{aligned}$$

où $\boldsymbol{X}_1$, $\boldsymbol{X}_2$,..., $\boldsymbol{X}_{p+1}$ *correspondent aux colonnes de* $\boldsymbol{X}$.

Ce système d'équations peut se réécrire sous la forme :

$$\boldsymbol{X}'(\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\theta}}) = 0 \text{ } i.e. \text{ } \boldsymbol{X}'\boldsymbol{X}\widehat{\boldsymbol{\theta}} = \boldsymbol{X}'\boldsymbol{Y}.$$

Nous donnons dans la proposition suivante les conditions sous lesquelles $\mathbf{X}'\mathbf{X}$ est inversible.

Proposition 5. *Soit* $\boldsymbol{X}$ *une matrice de taille* $n \times p$*, on a alors les propriétés suivantes :*

- *si* $p > n$, $\boldsymbol{X}'\boldsymbol{X}$ *n'est pas inversible.*
- *si* $p \leq n$, $\boldsymbol{X}'\boldsymbol{X}$ *est inversible si* $\boldsymbol{X}$ *est de rang plein i.e. si rang*$(\boldsymbol{X}) = p$*. Il s'agit d'une condition nécessaire et suffisante.*

Preuve de la proposition 5

Notons tout d'abord que $\mathbf{X}'\mathbf{X}$ est de taille $p \times p$.

Montrons que $\text{Ker}(\mathbf{X}'\mathbf{X}) = \text{Ker}(\mathbf{X})$. En effet, si $u \in \text{Ker}(\mathbf{X})$ alors $\mathbf{X}u = 0$ et donc $u \in \text{Ker}(\mathbf{X}'\mathbf{X})$. Réciproquement, si $u \in \text{Ker}(\mathbf{X}'\mathbf{X})$, on a donc :

$$(\mathbf{X}'\mathbf{X})u = 0 \text{ d'où l'on déduit que } u'(\mathbf{X}'\mathbf{X})u = 0 \text{ et donc } \|\mathbf{X}u\|^2 = 0,$$

ce qui implique que : $u \in \text{Ker}(\mathbf{X})$ d'où le résultat.

D'après le théorème du rang, $\text{rang}(\mathbf{X}'\mathbf{X}) + \dim(\text{Ker}(\mathbf{X}'\mathbf{X})) = p$ et $\text{rang}(\mathbf{X}) + \dim(\text{Ker}(\mathbf{X})) = p$. On a donc que $\text{rang}(\mathbf{X}'\mathbf{X}) = \text{rang}(\mathbf{X})$. De plus, $\text{rang}(\mathbf{X}) \leq \min(n, p)$.

Si $p > n$, $\text{rang}(\mathbf{X}) \leq n < p$ et donc $\text{Ker}(\mathbf{X}) \neq \{0\}$ d'où $\text{Ker}(\mathbf{X}'\mathbf{X}) \neq \{0\}$, ce qui implique que $\mathbf{X}'\mathbf{X}$ n'est pas inversible.

Si $p \leq n$ et que $\text{Ker}(\mathbf{X}) = \{0\}$ ce qui est équivalent à $\mathbf{X}$ est de rang plein alors $\text{Ker}(\mathbf{X}'\mathbf{X})) = \{0\}$ et $\text{rang}(\mathbf{X}'\mathbf{X}) = p$ donc $\mathbf{X}'\mathbf{X}$ est inversible.

Remarque 4. *La matrice* Π *définie par*

$$\Pi = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$$

est une matrice de projection orthogonale sur l'image de $\boldsymbol{X}$*. En effet,* $\forall x$, Πx *est dans l'image de* $\boldsymbol{X}$, Π *est symétrique car* $\Pi' = \Pi$ *et* $\Pi^2 = \Pi$ *(on dit que* Π *est idempotente).*

A l'aide de la théorie des vecteurs gaussiens (voir la section 3.5), on en déduit la loi de l'estimateur $\widehat{\boldsymbol{\theta}}$ et donc la loi des estimateurs $\hat{\theta}_k$.

Proposition 6. *Dans le modèle* (3.3)*, si* $\boldsymbol{X}$ *est de rang plein,*

$$\hat{\theta}_k \sim \mathcal{N}(\theta_k, \text{Cov}(\widehat{\boldsymbol{\theta}})_{k,k}),$$

où $\hat{\theta}_k$ *est la* k*-ième ligne du vecteur* $\widehat{\boldsymbol{\theta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$ *et* $\text{Cov}(\widehat{\boldsymbol{\theta}}) = \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1}$.

La preuve de la proposition est donnée dans la section 3.5.

3.3.2 Estimation de σ^2 dans (3.3)

En utilisant la méthode du maximum de vraisemblance, on obtient un estimateur de σ^2 en maximisant par rapport à σ^2 le critère ci-dessous :

$$-\frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - \widehat{\theta}_1 x_{i,1} - \cdots - \widehat{\theta}_p x_{i,p})^2.$$

En dérivant ce critère par rapport à σ^2, on obtient :

$$-\frac{n}{2} \times \frac{1}{\widehat{\sigma}^2_{\mathrm{MV}}} + \frac{1}{2\widehat{\sigma}^4_{\mathrm{MV}}}\sum_{i=1}^{n}(Y_i - \widehat{\theta}_1 x_{i,1} - \cdots - \widehat{\theta}_p x_{i,p})^2 = 0.$$

Ainsi l'estimateur de σ^2 obtenu par la méthode du maximum de vraisemblance est défini comme suit :

$$\widehat{\sigma}^2_{\mathrm{MV}} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{\theta}_1 x_{i,1} - \cdots - \widehat{\theta}_p x_{i,p})^2.$$

D'après le théorème de Cochran donné ci-dessous le défaut de cet estimateur est qu'il est biaisé, on lui préfére donc l'estimateur $\widehat{\sigma}^2$ qui, lui, est sans biais :

$$\widehat{\sigma}^2 = \frac{1}{n-p}\sum_{i=1}^{n}(Y_i - \widehat{\theta}_1 x_{i,1} - \cdots - \widehat{\theta}_p x_{i,p})^2.$$

Ce sera cet estimateur de σ^2 que l'on utilisera dans la suite.

Théorème 4 (Théorème de Cochran)**.** *Soient $Y_1, \ldots, Y_n$ des v.a. i.i.d. de loi $\mathcal{N}(0, \sigma^2)$. Soient V un sous-espace vectoriel de $\mathbb{R}^n$ de dimension r et $\Pi_V(\boldsymbol{Y})$ la projection orthogonale de Y sur V. Alors les variables aléatoires $\Pi_V(\boldsymbol{Y})$ et $\boldsymbol{Y} - \Pi_V(\boldsymbol{Y}) = (Id - \Pi_V)(\boldsymbol{Y}) = \Pi_{V^\perp}(\boldsymbol{Y})$ sont indépendantes. De plus, $\|\Pi_V(\boldsymbol{Y})\|^2/\sigma^2 \sim \chi^2(r)$ et $\|\boldsymbol{Y} - \Pi_V(\boldsymbol{Y})\|^2/\sigma^2 \sim \chi^2(n-r)$.*

La preuve du théorème de Cochran est donnée dans la section 3.6.

Par le théorème de Cochran, on peut montrer que :

$$\frac{(n-p)\widehat{\sigma}^2}{\sigma^2} \sim \chi^2(n-p).$$

3.3.3 Exemples

3.3.3.1 Exemple 1 : régression linéaire multiple

Dans le modèle de régression linéaire multiple défini par (3.5) où $\mathbf{X}$ a $p+1$ colonnes, on a d'après la proposition 5 que $\mathbf{X}'\mathbf{X}$ est inversible et que donc $\widehat{\boldsymbol{\theta}}$ est bien défini si $n \geq p+1$ et si les colonnes de $\mathbf{X}$ sont linéairement indépendantes (afin d'avoir $\mathbf{X}$ de rang plein).

3.3.3.2 Exemple 2 : ANOVA à 1 facteur

Dans le modèle d'ANOVA à 1 facteur défini par (3.7), la matrice $\mathbf{X}$ n'est pas de rang plein. En effet, ses colonnes ne sont pas linéairement indépendantes puisqu'on obtient la première en sommant les autres colonnes donc d'après la proposition 5, $\mathbf{X}'\mathbf{X}$ n'est pas inversible. Pour

résoudre le problème, il suffit de fixer un paramètre et donc de ne plus l'estimer et du coup de supprimer sa colonne correspondante dans $\mathbf{X}$.

Dans R, la contrainte usuelle est : $\alpha_1 = 0$. Avec cette contrainte, $\widehat{\boldsymbol{\theta}}$ peut être calculé et le nombre de paramètres "libres" du modèles *i.e.* à estimer est $p = I$ et non plus $I + 1$ comme dans le modèle initial.

3.3.4 Dans R

On peut obtenir l'estimation des paramètres du modèle à l'aide du logiciel R. Les valeurs des $\widehat{\theta}_k$ sont données dans la colonne `Estimate` de la sortie `summary`.

3.3.4.1 Exemple : chenille processionnaire du pin (régression linéaire multiple)

Dans le cas du modèle de régression linéaire multiple, la commande à taper est la suivante :

```
lm(Variable à expliquer  ~  Var. explicative 1 + ... + Var. explicative p)
```

Dans l'exemple des chenilles processionnaires du pin cela donne donc :

```
Nids.lm=lm(NbNids~Altitude+Pente+NbPins+Hauteur+Diametre+Densite+Orient
           +HautMax+NbStrat+Melange,data=chenilles)
summary(Nids.lm)
```

```
##
## Call:
## lm(formula = NbNids ~ Altitude + Pente + NbPins + Hauteur + Diametre +
##     Densite + Orient + HautMax + NbStrat + Melange, data = chenilles)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.03941 -0.26272 -0.02351  0.21953  1.35140
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.561849   2.096950   4.083 0.000493 ***
## Altitude    -0.002956   0.001038  -2.847 0.009374 **
## Pente       -0.034821   0.014510  -2.400 0.025311 *
## NbPins       0.035385   0.066586   0.531 0.600454
## Hauteur     -0.501564   0.378701  -1.324 0.198955
## Diametre     0.108739   0.069495   1.565 0.131925
## Densite     -0.032715   1.044915  -0.031 0.975305
## Orient      -0.203959   0.669598  -0.305 0.763535
## HautMax      0.028180   0.157007   0.179 0.859201
## NbStrat     -0.862409   0.572133  -1.507 0.145945
## Melange     -0.448124   0.513764  -0.872 0.392499
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.5493 on 22 degrees of freedom
## Multiple R-squared:  0.6809, Adjusted R-squared:  0.5359
## F-statistic: 4.695 on 10 and 22 DF,  p-value: 0.001203
```

On lit dans le tableau que :

— $\widehat{\beta}_0 = 8.561849$
— $\widehat{\beta}_1$ = -0.002956
— $\widehat{\beta}_2$ = -0.034821
— $\vdots$

La valeur de $\widehat{\sigma}$ est donnée après le tableau sur la ligne `Residual standard error`. Ici $\widehat{\sigma}^2 = 0.5493^2 = 0.3017$ et 22 correspond à $n - (p+1) = 33 - 11$ où $p+1$ correspond au nombre de paramètres (β_0, β_1,..., β_p) dans le modèle de régression linéaire multiple.

3.3.4.2 Exemple : influence du genre d'un individu sur sa taille (ANOVA 1 facteur)

La commande à taper dans R pour estimer les paramètres de l'ANOVA à 1 facteur est la suivante.

```
taille.lm=lm(taille~genre,data=PT)
summary(taille.lm)
```

```
##
## Call:
## lm(formula = taille ~ genre, data = PT)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.272 -2.097  0.019  1.923  4.863
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  65.1415     0.4492  145.00  < 2e-16 ***
## genrehomme    6.1071     0.5965   10.24 3.47e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.419 on 65 degrees of freedom
## Multiple R-squared:  0.6172, Adjusted R-squared:  0.6113
## F-statistic: 104.8 on 1 and 65 DF,  p-value: 3.469e-15
```

Comme précédemment on lit l'estimation des paramètres μ et α_i dans la colonne "Estimate" du tableau. Plus précisément,

— $\widehat{\mu} = 65.141537$
— $\widehat{\alpha}_2 = \widehat{\alpha}_{\text{homme}} = 6.107095$

A cause de la contrainte $\alpha_1 = 0$ qui correspond ici à $\alpha_{\text{femme}} = 0$ car la modalité "femme" est

avant la modalité “homme” dans l'ordre alphabétique, $\widehat{\alpha}_{\text{femme}} = \widehat{\alpha}_1 = 0$, elle n'apparaît donc pas dans le tableau.

La valeur de $\widehat{\sigma}$ est donnée après le tableau sur la ligne `Residual standard error`. Ici $\widehat{\sigma}^2 = 2.419^2 = 5.8516$ et `65` correspond à $n - I = 67 - 2$ où I correspond au nombre de paramètres libres à estimer du modèle (μ, α_2).

3.4 Validation du modèle

On propose ci-dessous plusieurs manières de valider un modèle linéaire que l'on a ajusté à des données.

3.4.1 Graphes de diagnostic

Le modèle linéaire suppose que :

1. la variance du terme d'erreur est constante (modèle dit homoscédastique)
2. les erreurs sont indépendantes
3. les erreurs sont gaussiennes

La validation de ces hypothèses se fait en regardant le graphe des résidus définis par :

$$\widehat{E}_i = Y_i - \widehat{Y}_i \quad \text{où } \widehat{Y}_i = \widehat{\theta}_1 x_{i,1} + \cdots + \widehat{\theta}_p x_{i,p}.$$

De façon générale, une structure particulière du nuage de points du graphe des résidus représentant les points de coordonnées $(\widehat{Y}_i, \widehat{E}_i)$ indique que le modèle proposé n'est pas adapté aux données.

- Si 1. n'est pas vérifiée, les deux transformations les plus courantes sont : $\log(Y_i)$ et $\sqrt{Y_i}$. On pourra plus généralement utiliser une transformation de Box-Cox.
- On ne teste pas 2. et en général, on fait confiance à la façon dont les expériences ont été menées.
- Pour vérifier 3., on peut utiliser un Q-Q plot ou des tests statistiques comme le test de Kolmogorov-Smirnov ou le test de Shapiro-Wilk.
- On peut aussi vérifier la présence de points aberrants ou "outliers" en anglais.

Le logiciel R fournit 4 graphes de diagnostic :

- Le premier graphique, en haut à gauche, représente le graphe des résidus et ne doit présenter aucune structure particulière.
- Le deuxième graphique, en haut à droite, est le Q-Q plot entre les résidus normalisés et la loi normale $\mathcal{N}(0, 1)$ et permet de vérifier l'hyptothèse de normalité.
- Le troisième graphique, en bas à gauche, permet de valider l'homoscédasticité du modèle.
- Le quatrième graphique, en bas à droite, permet de détecter la présence de points aberrants.

On représente dans la figure 3.2 les graphes de diagnostic fournis par R que l'on obtient lorsque l'on ajuste un modèle de régression linéaire simple à des données obéissant à un modèle de régression linéaire simple *i.e.* lorsque l'on ne commet pas d'erreur de modélisation.

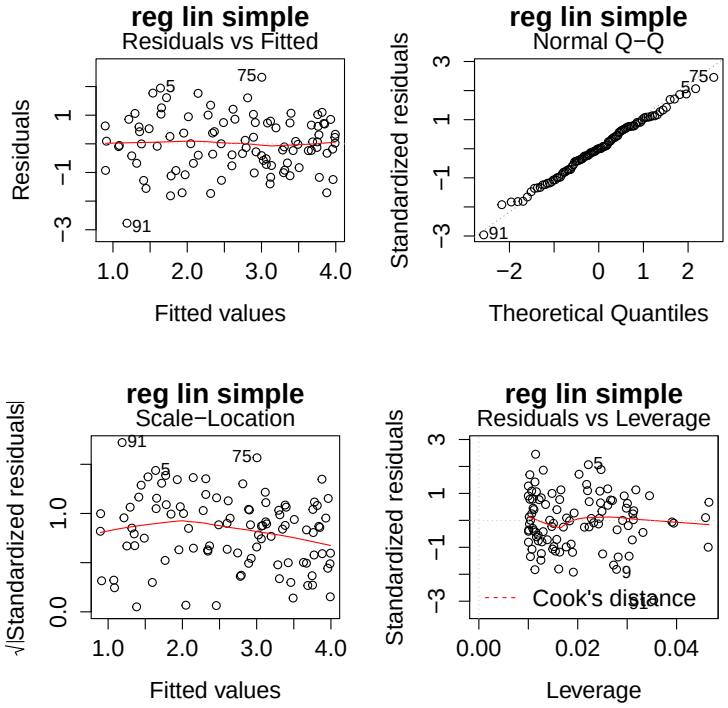


FIGURE 3.2 – Modèle : $Y_i = 1 + 3x_i + E_i,\ 1 \leq i \leq 100$

Lorsque l'on applique un modèle de régression linéaire simple à des données obéissant à un modèle de régression linéaire simple à variance non constante (hétéroscédastique) on obtient les graphes de diagnostic de la figure 3.3.

Lorsque l'on ajuste un modéle de régression linéaire simple à des données obéissant à un modèle ayant une variable supplémentaire ici x^2 on obtient les graphes de la figure 3.4.

Enfin on représente dans la figure 3.5 les graphes de diagnostic que l'on obtient en présence de valeurs aberrantes dans les données.

3.4.2 Critère du R^2

Les moyens complémentaires dont on dispose pour valider un modèle sont les critères du R^2 et du R^2 ajusté.

Définition 13 (Définition du R^2)**.**

$$R^2 = \frac{SCM}{SCT},$$

où

$$\begin{aligned} SCT &= \sum_{i=1}^{n}(Y_i - \overline{Y})^2 \\ SCM &= \sum_{i=1}^{n}(\widehat{Y}_i - \overline{Y})^2 \ \textit{avec}\ \overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i \ \textit{et}\ \widehat{Y}_i \ \textit{la prédiction faite par le modèle.} \end{aligned}$$

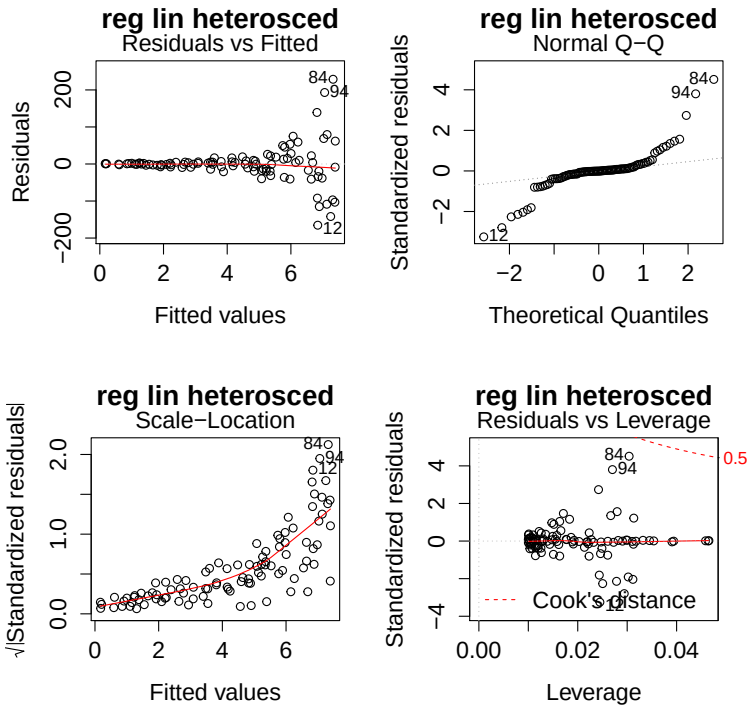


FIGURE 3.3 – Modèle : $Y_i = 1 + 3x_i + \sigma_i E_i,\ 1 \leq i \leq 100$

On a que :

$$SCT = SCM + SCR,$$

où

$$SCR = \sum_{i=1}^{n} (Y_i - \widehat{Y}_i)^2.$$

Les quantités SCT, SCM et SCR représentent respectivement la variabilité totale, la variabilité due au modèle et la variabilité résiduelle des données. Un modèle est d'autant meilleur que la quantité SCR est petite (*i.e.* la quantité SCM est grande).

On en déduit donc une autre définition du R^2 :

$$R^2 = 1 - \frac{SCR}{SCT}.$$

Plus le R^2 est proche de 1 et meilleur est le modèle.

On peut définir également le $R^2_{\text{ajusté}}$.

Définition 14 (Définition du $R^2_{\text{ajusté}}$ ou R^2_{adjusted})**.**

$$R^2_{adj} = 1 - \frac{SCR/(n-p)}{SCT/(n-1)},$$

où p correspond au nombre de paramètres θ_1,...,θ_p du modèle linéaire.

Le logiciel R fournit les valeurs du R^2 et du R^2 ajusté comme une sortie de la commande `lm`. Ce sont respectivement les valeurs de `Multiple R-squared` et de `Adjusted R-squared`.

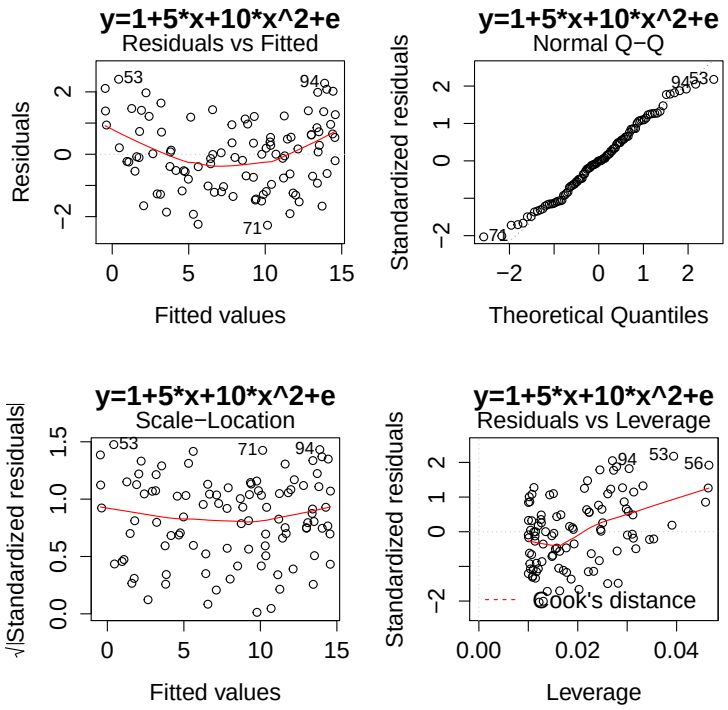


FIGURE 3.4 – Modèle : $Y_i = 1 + 5x_i + 10x_i^2 + E_i,\ 1 \leq i \leq 100$

3.4.3 Test du modèle

Dans les modèles linéaires, on a usuellement $x_{i,1} = 1$ pour tout i, ce qui signifie que la 1ère colonne de $\mathbf{X}$ ne contient que des 1. Le paramètre θ_1 représente alors une moyenne commune à toutes les observations indépendamment des effets spécifiques éventuels des variables explicatives.

On souhaite savoir si au moins une des autres variables explicatives $(x_2, \ldots, x_p)$ est pertinente pour expliquer la variable d'intérêt. On teste donc :

$$(H_0) : \text{"le modèle est } Y_i = \theta_1 + E_i,\ 1 \leq i \leq n\text{"} \quad \text{contre}$$

$$(H_1) : \text{"le modèle est } Y_i = \theta_1 + \theta_2 x_{i,2} + \cdots + \theta_p x_{i,p} + E_i,\ 1 \leq i \leq n$$
$$\text{où au moins l'un des } \theta_2, \ldots, \theta_p \text{ est non nul"},$$

ce qui peut encore être réécrit comme suit :

$$(H_0) : \text{"}\forall i \geq 2,\ \theta_i = 0\text{"} \text{ contre } (H_1) : \text{"}\exists i \geq 2,\ \theta_i \neq 0\text{"}.$$

Si on retient (H_0), cela signifie qu'aucune des variables explicatives utilisées n'est pertinente pour expliquer Y.

De façon générale, plus on considère un modèle M_1 "compliqué'' *i.e.* comprenant plus de paramètres qu'un modèle M_0, plus l'ajustement est en général meilleur et donc : $SCR_1 \leq SCR_0$ où SCR_1 est le SCR sous (H_1) et SCR_0 est le SCR sous (H_0). On rejettera donc (H_0) si le modèle M_0 est suffisamment différent du modèle M_1 *i.e* si $SCR_0 - SCR_1$ dépasse un certain

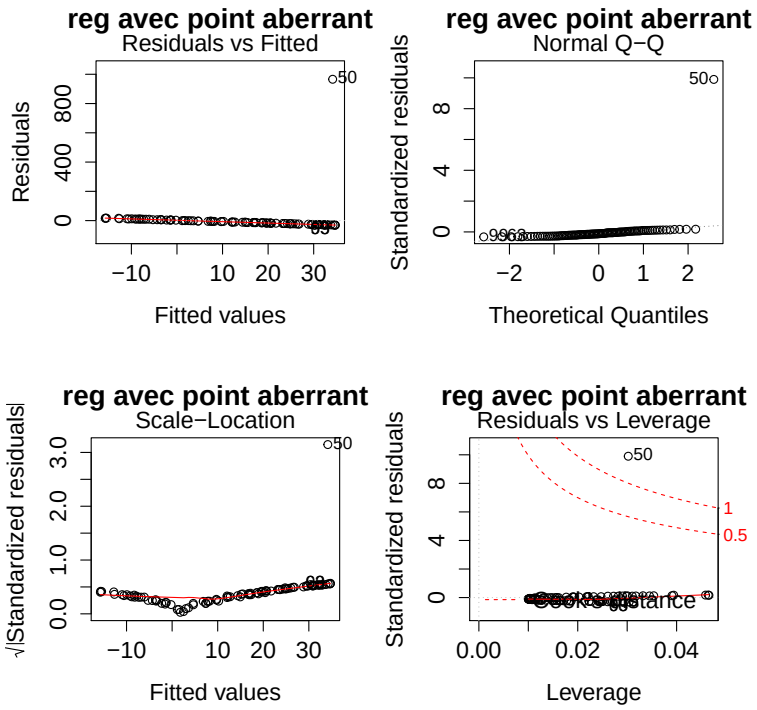


FIGURE 3.5 – Modèle : $Y_i = 1 + 3x_i + E_i,\ 1 \leq i \leq 100,\ i \neq 50$ et $Y_{50} = 1000$

seuil mais on doit être sûr de proposer une statistique libre *i.e.* indépendante de tout paramètre inconnu sous (H_0).

On propose donc la statistique suivante :

$$F = \frac{(SCR_0 - SCR_1)/(p-1)}{SCR_1/(n-p)} = \frac{SCM/(p-1)}{SCR/(n-p)}, \tag{3.10}$$

où :

$$SCR_0 - SCR_1 = \sum_{i=1}^{n} (\widehat{Y}_i - \overline{Y})^2 = SCM \text{ et } SCR = \sum_{i=1}^{n} (Y_i - \widehat{Y}_i)^2.$$

Proposition 7. *Sous (H_0), F suit une loi de Fisher à $p-1$ et $n-p$ degrés de liberté notée $\mathcal{F}(p-1, n-p)$.*

On rejettera donc l'hypothèse (H_0) au niveau α si

$$F > s_\alpha, \text{ où } s_\alpha \text{ est tel que } \mathbb{P}_{H_0}(F > s_\alpha) = \mathbb{P}(\mathcal{F} > s_\alpha) \leq \alpha,$$

où $\mathcal{F}$ suit une loi de Fisher à $p-1$ et $n-p$ degrés de liberté.

Preuve de la Proposition 7 Soient V le sous-espace vectoriel (sev) engendré par les colonnes de X et W le sev engendré par le vecteur $(1, 1, \ldots, 1)'$.

On a donc : $(\widehat{Y}_i - \overline{Y})_{1 \leq i \leq n} = \Pi_{V \cap W^\perp}(Y)$, où $\Pi_{V \cap W^\perp}$ désigne la projection orthogonale sur $V \cap W^\perp$. De plus, $(Y_i - \widehat{Y}_i)_{1 \leq i \leq n} = \Pi_{V^\perp}(Y)$. En utilisant le théorème de Cochran énoncé et prouvé ci-dessous : $(SCR_0 - SCR_1)/\sigma^2 \overset{(H_0)}{\hookrightarrow} \chi^2(p-1)$ et $SCR_1/\sigma^2 \overset{(H_0)}{\hookrightarrow} \chi^2(n-p)$.

De plus, $SCR_0 - SCR_1$ et SCR_1 sont indépendantes et F suit donc une loi de Fisher à $p-1$ et $n-p$ degrés de liberté.

On remarque que

$$R^2_{\text{Adj}} = 1 - \frac{SCR_1/(n-p)}{SCR_0/(n-1)} = 1 - \frac{\widehat{\sigma}^2_1}{\widehat{\sigma}^2_0},$$

où $\widehat{\sigma}^2_0$ et $\widehat{\sigma}^2_1$ correspondent aux estimateurs de σ^2 sous (H_0) et (H_1) respectivement.

3.4.4 Dans R

3.4.4.1 Exemple : chenille processionnaire du pin (régression linéaire multiple)

On lit la p-valeur associée au test du modèle dans la dernière ligne de l'affichage de `summary(Nids.lm)`.

```
Nids.lm=lm(NbNids~Altitude+Pente+NbPins+Hauteur+Diametre+Densite+Orient
           +HautMax+NbStrat+Melange,data=chenilles)
summary(Nids.lm)
```

```
##
## Call:
## lm(formula = NbNids ~ Altitude + Pente + NbPins + Hauteur + Diametre +
##     Densite + Orient + HautMax + NbStrat + Melange, data = chenilles)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.03941 -0.26272 -0.02351  0.21953  1.35140
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.561849   2.096950   4.083 0.000493 ***
## Altitude    -0.002956   0.001038  -2.847 0.009374 **
## Pente       -0.034821   0.014510  -2.400 0.025311 *
## NbPins       0.035385   0.066586   0.531 0.600454
## Hauteur     -0.501564   0.378701  -1.324 0.198955
## Diametre     0.108739   0.069495   1.565 0.131925
## Densite     -0.032715   1.044915  -0.031 0.975305
## Orient      -0.203959   0.669598  -0.305 0.763535
## HautMax      0.028180   0.157007   0.179 0.859201
## NbStrat     -0.862409   0.572133  -1.507 0.145945
## Melange     -0.448124   0.513764  -0.872 0.392499
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5493 on 22 degrees of freedom
## Multiple R-squared:  0.6809, Adjusted R-squared:  0.5359
## F-statistic: 4.695 on 10 and 22 DF,  p-value: 0.001203
```

La p-valeur vaut ici : 0.001203.

De plus, $R^2 = 0.6809$ et $R^2_{\text{Adj}} = 0.5359$.

On a accès aux valeurs de SCM et SCR intervenant dans (3.10) en tapant la commande suivante :

```
anova(Nids.lm)
```

```
## Analysis of Variance Table
##
## Response: NbNids
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Altitude   1 5.8476  5.8476 19.3835 0.0002256 ***
## Pente      1 3.2360  3.2360 10.7266 0.0034599 **
## NbPins     1 1.0086  1.0086  3.3432 0.0810752 .
## Hauteur    1 0.2040  0.2040  0.6764 0.4196709
## Diametre   1 2.4208  2.4208  8.0246 0.0096860 **
## Densite    1 0.0814  0.0814  0.2697 0.6087268
## Orient     1 0.0185  0.0185  0.0614 0.8065795
## HautMax    1 0.1681  0.1681  0.5573 0.4632390
## NbStrat    1 0.9487  0.9487  3.1446 0.0900245 .
## Melange    1 0.2295  0.2295  0.7608 0.3924994
## Residuals 22 6.6369  0.3017
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

On lit que $SCR = 6.63$ et on déduit que $SCM = 5.8476 + 3.2360 + \cdots + 0.2295 = 14.163$. Ainsi, $SCT = SCM + SCR = 20.8002$. On retrouve aussi ces SC avec la commande suivante qui réalise le test du modèle :

```
Nids.lm.0=lm(NbNids~1,data=chenilles)
anova(Nids.lm.0,Nids.lm)
```

```
## Analysis of Variance Table
##
## Model 1: NbNids ~ 1
## Model 2: NbNids ~ Altitude + Pente + NbPins + Hauteur + Diametre + Densite +
##     Orient + HautMax + NbStrat + Melange
##   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
## 1     32 20.8002
## 2     22  6.6369 10    14.163 4.6948 0.001203 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.4.4.2 Exemple : influence du genre d'un individu sur sa taille (ANOVA 1 facteur)

```
taille.lm=lm(taille~genre,data=PT)
summary(taille.lm)
```

```
##
## Call:
## lm(formula = taille ~ genre, data = PT)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.272 -2.097  0.019  1.923  4.863
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  65.1415     0.4492  145.00  < 2e-16 ***
## genrehomme    6.1071     0.5965   10.24 3.47e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.419 on 65 degrees of freedom
## Multiple R-squared:  0.6172, Adjusted R-squared:  0.6113
## F-statistic: 104.8 on 1 and 65 DF,  p-value: 3.469e-15
```

Sur la ligne F-statistic, on lit la p-valeur du test du modèle qui vaut $3.469 \times 10^{-15} < 0.05$ donc on rejette (H_0) au niveau 5% ce qui signifie que le genre a une influence sur la taille.

Les deux sections suivantes apportent des explications théoriques sur les propriétés des estimateurs des paramètres d'un modèle linéaire et ne sont pas au programme de ce cours.

3.5 Introduction aux vecteurs gaussiens

Définition 15 (Vecteur gaussien)**.** *Un vecteur aléatoire Z à valeurs dans $\mathbb{R}^d$ est dit **gaussien** si toute combinaison linéaire de ses composantes suit une loi normale.*

Remarque 5. *Si $Z_1, \ldots, Z_d$ sont des variables gaussiennes indépendantes, alors $Z = (Z_1, \ldots, Z_d)'$ est un vecteur gaussien.*

Remarque 6. *Les composantes d'un vecteur gaussien sont des variables aléatoires gaussiennes. La réciproque est fausse.*

Comme pour la loi gaussienne unidimensionnelle, un vecteur gaussien Z est entièrement caractérisé par son espérance $m = \mathbb{E}(Z)$ et sa matrice de covariance $\Sigma = \mathrm{Cov}(Z)$ et on note $Z \sim \mathcal{N}_d(m, \Sigma)$.

Propriétés 2. *Soit $Z = (Z_1, \ldots, Z_d)' \sim \mathcal{N}_d(m, \Sigma)$.*

- *Pour toute matrice A de taille $q \times d$ et pour tout vecteur $b \in \mathbb{R}^q$,*

$$AZ + b \sim \mathcal{N}_q(Am + b, A\Sigma A').$$

- *Pour tous $i \neq j$, on a : Z_i et Z_j sont indépendantes $\quad\Leftrightarrow\quad \mathrm{Cov}(Z_i, Z_j) = 0$.*

Théorème 5. *Soit $Z = (Z_1, \ldots, Z_d)' \sim \mathcal{N}_d(m, \Sigma)$. La loi de Z admet une densité $f_{m,\Sigma}$ par rapport à la mesure de Lebesgue sur $\mathbb{R}^d$ si et seulement si Σ est inversible. Dans ce cas,*

$$\forall z \in \mathbb{R}^d, f_{m,\Sigma}(z) = \left(\frac{1}{\sqrt{2\pi}}\right)^d \frac{1}{\sqrt{\mathrm{Det}(\Sigma)}} \exp\left(-\frac{1}{2}(z-m)'\Sigma^{-1}(z-m)\right).$$

Preuve de la proposition 6

Dans le modèle (3.3), les E_i sont i.i.d.,de loi $\mathcal{N}(0, \sigma^2)$. Ainsi, $\mathbf{E} \sim \mathcal{N}_n(0, \sigma^2 I_n)$ et par propriété de linéarité des vecteurs gaussiens, $\mathbf{Y} \sim \mathcal{N}_n(\mathbf{X}\boldsymbol{\theta}, \sigma^2 I_n)$. Comme $\widehat{\boldsymbol{\theta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$, on a également que $\widehat{\boldsymbol{\theta}}$ est un vecteur gaussien $\mathcal{N}_p(\boldsymbol{\theta}, \sigma^2(\mathbf{X}'\mathbf{X})^{-1})$ et on en extrait aisément la loi des $\hat{\theta}_k$.

3.6 Preuve du théorème de Cochran

Lemme 1. *Soient $u = (u_1, \ldots, u_n)$ un vecteur unitaire de $\mathbb{R}$ i.e. tel que : $\|u\|^2 = \sum_{i=1}^n u_i^2 = 1$ et $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ le vecteur aléatoire tel que les Y_i sont i.i.d. de loi $\mathcal{N}(0,1)$. La projection orthogonale de $\boldsymbol{Y}$ sur u s'écrit alors :*

$$\Pi_u(\boldsymbol{Y}) = c_u(\boldsymbol{Y})\, u =< \boldsymbol{Y}, u > \, u,$$

où $< u, w >$ désigne le produit scalaire entre 2 vecteurs $u = (u_1, \ldots, u_n)$ et $w = (w_1, \ldots, w_n)$ de $\mathbb{R}^n$ défini par $< u, w >= \sum_{i=1}^n u_i w_i$. De plus, $c_u(\boldsymbol{Y}) \sim \mathcal{N}(0,1)$.

Preuve du Lemme 1 Par définition, Π_u est tel que : $< \mathbf{Y} - \Pi_u(\mathbf{Y}), u >= 0$ où $\Pi_u(\mathbf{Y}) = c\, u$. On a donc : $< \mathbf{Y}, u > -c < u, u >= 0$ d'où l'on déduit que $c =< \mathbf{Y}, u >$ puisque $\|u\|^2 = 1$. Par ailleurs, $c_u(\mathbf{Y}) = \sum_{i=1}^n u_i Y_i$ donc $c_u(\mathbf{Y})$ est une gaussienne centrée de variance $\sum_{i=1}^n u_i^2 = 1$, ce qui conclut la preuve du lemme.

Lemme 2. *Soient u et w deux vecteurs de $\mathbb{R}^n$, $c_u(\boldsymbol{Y}) =< \boldsymbol{Y}, u >$ et $c_w(\boldsymbol{Y}) =< \boldsymbol{Y}, w >$, où $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ est le vecteur aléatoire tel que les Y_i sont i.i.d. de loi $\mathcal{N}(0,1)$. Alors $c_u(\boldsymbol{Y})$ et $c_w(\boldsymbol{Y}) = 0$ sont indépendantes si et seulement si $< u, w >= 0$.*

Preuve du Lemme 2 Comme les Y_i sont i.i.d. de loi $\mathcal{N}(0,1)$,

$$\mathrm{Cov}(c_u(\mathbf{Y}), c_w(\mathbf{Y})) = \mathbb{E}(c_u(\mathbf{Y})\, c_w(\mathbf{Y})) = \mathbb{E}\left[\left(\sum_{i=1}^n u_i Y_i\right)\left(\sum_{j=1}^n w_j Y_j\right)\right]$$

$$= \sum_{1 \le i,j \le n} u_i w_j \mathbb{E}(Y_i Y_j) = \sum_{i=1}^n u_i w_i =< u, w > .$$

Etant donné que $c_u(\mathbf{Y})$ et $c_w(\mathbf{Y})$ sont gaussiennes, elles sont indépendantes si et seulement si elles sont décorrélées et donc d'après le résultat ci-dessus si et seulement si $< u, w >= 0$.

Preuve du théoèreme de Cochran Quitte à considérer Y_i/σ à la place de Y_i on peut supposer que les Y_i sont $\mathcal{N}(0,1)$. On construit $(v_1, \ldots, v_r)$ une base orthonormée de V (c'est possible grâce au principe d'orthogonalisation de Gram-Schmidt). On a donc :

$$\Pi_V(\mathbf{Y}) = \sum_{i=1}^r c_{v_i}(\mathbf{Y}) v_i$$

et $\|\Pi_V(\mathbf{Y})\|^2 = \sum_{i=1}^{r} c_{v_i}(\mathbf{Y})^2$. D'où l'on déduit d'après le lemme 1 que $\|\Pi_V(\mathbf{Y})\|^2 \sim \chi^2(r)$.

En complétant la base $v_1, \ldots, v_r$ en $v_{r+1}, \ldots, v_n$, cette dernière constitue une base orthonormée de $V^\perp$. On a donc de la même façon que :

$$\Pi_{V^\perp}(\mathbf{Y}) = \sum_{i=r+1}^{n} c_{v_i}(\mathbf{Y}) v_i$$

et $\|\Pi_{V^\perp}(\mathbf{Y})\|^2 = \sum_{i=r+1}^{n} c_{v_i}(\mathbf{Y})^2$. D'où l'on déduit d'après le Lemme 1 que $\|\Pi_{V^\perp}(\mathbf{Y})\|^2 \sim \chi^2(n-r)$.

On déduit de plus du Lemme 2 que $\Pi_V(\mathbf{Y})$ et $\Pi_{V^\perp}(\mathbf{Y})$ sont indépendantes puisque les v_i sont orthogonaux deux à deux.

Chapitre 4

Exemples classiques de modèles linéaires

4.1 Compléments pour le modèle de régression linéaire multiple : sélection de variables

On reprend l'exemple des chenilles processionnaires introduit à la section 3.2.2.1 pour lequel on a proposé un modèle de régression linéaire multiple :

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

avec

- Y_i la variable aléatoire modélisant le nombre de nids dans la parcelle i, i allant de 1 à $n = 33$,
- $x_{i,k}$ la valeur de la k-ième variable pour la parcelle i, k allant de 1 à $p = 10$, et
- les E_i qui sont i.i.d. de loi $\mathcal{N}(0, \sigma^2)$.

Le but est de proposer le modèle le plus simple possible *i.e.* comportant le moins possible de variables tout en s'ajustant le mieux possible aux données. Si certaines variables explicatives sont très corrélées entre elles, les garder toutes produirait une redondance d'information et c'est la raison pour laquelle on souhaite les éliminer.

Si on devait tester toutes les configurations possibles pour chacun des β_i *i.e.* $\beta_i = 0$ ou $\beta_i \neq 0$, on aurait 2^{p+1} modèles à tester (puisque 2 possibilités pour chaque paramètre) ce qui serait trop coûteux en temps de calcul même pour p petit.

L'idée est plutôt d'utiliser des méthodes pas à pas. Il en existe plusieurs :

- des procédures "ascendantes" où on part du plus petit modèle et on l'enrichit,
- des procédures "descendantes" où on part du modèle le plus compliqué que l'on simplifie.

On peut utiliser le critère d'information d'Akaike (AIC : Akaike Information Criterion) qui est disponible dans R et que l'on cherche à minimiser. C'est le critère qui est utilisé dans la fonction `step` de R qui fournit les affichages ci-dessous.

L'AIC d'un modèle M se définit par

$$AIC(M) = -2 \log L_M(y; \hat{\boldsymbol{\theta}}, \hat{\sigma}^2) + 2D_M$$

où $L_M(y; \hat{\boldsymbol{\theta}}, \hat{\sigma}^2)$ est la vraisemblance du modèle M et D_M est le nombre de paramètres à estimer du modèle M. Ce critère repose donc sur un compromis entre la qualité de l'ajustement et la complexité du modèle.

On part du modèle complet et on observe que la valeur de l'AIC est minimisée à la 1ère étape lorsque l'on enlève la variable `Densite` donc on l'enlève et on continue. A l'étape suivante c'est le fait d'enlever la variable `HautMax` qui minimise l'AIC donc on l'enlève et on continue jusqu'à ce que le fait d'enlever des variables ne fasse plus diminuer la valeur de l'AIC (`<none>`). Ici le modèle final est celui où l'on ne retient que les variables : `Altitude`, `Pente`, `Hauteur`, `Diametre` et `NbStrat`.

```
step(Nids.lm)
```

```
## Start:  AIC=-30.93
## NbNids ~ Altitude + Pente + NbPins + Hauteur + Diametre + Densite +
##     Orient + HautMax + NbStrat + Melange
##
##             Df Sum of Sq    RSS     AIC
## - Densite    1   0.00030 6.6372 -32.926
## - HautMax    1   0.00972 6.6466 -32.879
## - Orient     1   0.02799 6.6649 -32.788
## - NbPins     1   0.08520 6.7221 -32.506
## - Melange    1   0.22952 6.8664 -31.805
## <none>                   6.6369 -30.927
## - Hauteur    1   0.52918 7.1661 -30.396
## - NbStrat    1   0.68545 7.3224 -29.684
## - Diametre   1   0.73859 7.3755 -29.445
## - Pente      1   1.73726 8.3742 -25.255
## - Altitude   1   2.44545 9.0824 -22.576
##
## Step:  AIC=-32.93
## NbNids ~ Altitude + Pente + NbPins + Hauteur + Diametre + Orient +
##     HautMax + NbStrat + Melange
##
##             Df Sum of Sq    RSS     AIC
## - HautMax    1   0.01018 6.6474 -34.875
## - Orient     1   0.03630 6.6735 -34.746
## - Melange    1   0.29401 6.9312 -33.496
## <none>                   6.6372 -32.926
## - NbPins     1   0.49683 7.1340 -32.544
## - Hauteur    1   0.64047 7.2777 -31.886
## - NbStrat    1   0.77003 7.4073 -31.304
## - Diametre   1   0.81101 7.4482 -31.122
## - Pente      1   1.74141 8.3786 -27.237
## - Altitude   1   2.46888 9.1061 -24.490
```

```
##
## Step:  AIC=-34.88
## NbNids ~ Altitude + Pente + NbPins + Hauteur + Diametre + Orient +
##     NbStrat + Melange
##
##            Df Sum of Sq    RSS     AIC
## - Orient    1   0.03677 6.6842 -36.693
## - Melange   1   0.31511 6.9625 -35.347
## <none>                  6.6474 -34.875
## - NbPins    1   0.52769 7.1751 -34.354
## - Hauteur   1   0.75554 7.4029 -33.323
## - Diametre  1   0.80235 7.4498 -33.115
## - NbStrat   1   1.05891 7.7063 -31.997
## - Pente     1   1.75080 8.3982 -29.160
## - Altitude  1   2.57672 9.2241 -26.065
##
## Step:  AIC=-36.69
## NbNids ~ Altitude + Pente + NbPins + Hauteur + Diametre + NbStrat +
##     Melange
##
##            Df Sum of Sq    RSS     AIC
## - Melange   1   0.40168 7.0859 -36.767
## <none>                  6.6842 -36.693
## - NbPins    1   0.50781 7.1920 -36.277
## - NbStrat   1   1.02215 7.7063 -33.997
## - Hauteur   1   1.03389 7.7181 -33.947
## - Diametre  1   1.12553 7.8097 -33.558
## - Pente     1   1.71868 8.4029 -31.142
## - Altitude  1   2.98918 9.6734 -26.495
##
## Step:  AIC=-36.77
## NbNids ~ Altitude + Pente + NbPins + Hauteur + Diametre + NbStrat
##
##            Df Sum of Sq    RSS     AIC
## - NbPins    1   0.35221 7.4381 -37.167
## <none>                  7.0859 -36.767
## - Hauteur   1   0.85056 7.9364 -35.027
## - Diametre  1   0.99324 8.0791 -34.439
## - NbStrat   1   0.99727 8.0831 -34.422
## - Pente     1   1.82065 8.9065 -31.221
## - Altitude  1   2.62466 9.7105 -28.369
##
## Step:  AIC=-37.17
## NbNids ~ Altitude + Pente + Hauteur + Diametre + NbStrat
##
##            Df Sum of Sq    RSS     AIC
## <none>                  7.4381 -37.167
```

```
## - NbStrat   1   0.85317 8.2912 -35.583
## - Hauteur   1   1.21834 8.6564 -34.161
## - Diametre  1   1.37527 8.8133 -33.568
## - Pente     1   1.72426 9.1623 -32.286
## - Altitude  1   2.32266 9.7607 -30.199
```

```
##
## Call:
## lm(formula = NbNids ~ Altitude + Pente + Hauteur + Diametre +
##     NbStrat, data = chenilles)
##
## Coefficients:
## (Intercept)     Altitude        Pente      Hauteur     Diametre      NbStrat
##    5.998179    -0.002292    -0.033809    -0.521596     0.124145    -0.384935
```

On peut aussi procéder comme suit : partir du modèle dans lequel on ne met qu'une constante et rajouter les autres variables au fur et à mesure.

```
reg0=lm(NbNids~1,data=chenilles)
select.variables=step(reg0,scope=~Altitude+Pente+NbPins+Hauteur+Diametre
                +Densite+Orient+HautMax+NbStrat+Melange,direction="forward")
```

```
## Start:  AIC=-13.23
## NbNids ~ 1
##
##             Df Sum of Sq    RSS     AIC
## + NbStrat   1    8.4102 12.390 -28.328
## + Densite   1    6.7637 14.037 -24.210
## + NbPins    1    6.6141 14.186 -23.860
## + HautMax   1    6.3180 14.482 -23.178
## + Altitude  1    5.8476 14.953 -22.124
## + Pente     1    4.3148 16.485 -18.903
## + Hauteur   1    2.6644 18.136 -15.755
## <none>                  20.800 -13.231
## + Orient    1    0.9326 19.867 -12.745
## + Diametre  1    0.5178 20.282 -12.063
## + Melange   1    0.2645 20.536 -11.653
##
## Step:  AIC=-28.33
## NbNids ~ NbStrat
##
##             Df Sum of Sq    RSS     AIC
## + Altitude  1   2.14217 10.248 -32.592
## + Pente     1   1.43537 10.955 -30.391
## <none>                  12.390 -28.327
## + Orient    1   0.61447 11.775 -28.006
## + Hauteur   1   0.11369 12.276 -26.632
## + Melange   1   0.00749 12.383 -26.347
```

```
## + Densite   1   0.00665 12.383 -26.345
## + HautMax   1   0.00532 12.385 -26.342
## + NbPins    1   0.00366 12.386 -26.337
## + Diametre  1   0.00339 12.387 -26.337
##
## Step:  AIC=-32.59
## NbNids ~ NbStrat + Altitude
##
##            Df Sum of Sq     RSS     AIC
## + Pente     1   1.42717  8.8206 -35.541
## + Densite   1   0.77259  9.4752 -33.178
## <none>                  10.2478 -32.592
## + NbPins    1   0.56147  9.6863 -32.451
## + Orient    1   0.16975 10.0780 -31.143
## + Melange   1   0.14609 10.1017 -31.065
## + Diametre  1   0.13720 10.1106 -31.036
## + HautMax   1   0.00597 10.2418 -30.611
## + Hauteur   1   0.00437 10.2434 -30.606
##
## Step:  AIC=-35.54
## NbNids ~ NbStrat + Altitude + Pente
##
##            Df Sum of Sq    RSS     AIC
## + Densite   1   0.79916 8.0215 -36.675
## + NbPins    1   0.74105 8.0796 -36.436
## <none>                  8.8206 -35.541
## + Orient    1   0.42454 8.3961 -35.168
## + Diametre  1   0.16423 8.6564 -34.161
## + Melange   1   0.07765 8.7430 -33.832
## + Hauteur   1   0.00730 8.8133 -33.568
## + HautMax   1   0.00143 8.8192 -33.546
##
## Step:  AIC=-36.67
## NbNids ~ NbStrat + Altitude + Pente + Densite
##
##            Df Sum of Sq    RSS     AIC
## + Orient    1   0.58518 7.4363 -37.174
## <none>                  8.0215 -36.675
## + Diametre  1   0.11932 7.9022 -35.169
## + Melange   1   0.09567 7.9258 -35.071
## + NbPins    1   0.02181 7.9997 -34.765
## + HautMax   1   0.00583 8.0157 -34.699
## + Hauteur   1   0.00305 8.0184 -34.687
##
## Step:  AIC=-37.17
## NbNids ~ NbStrat + Altitude + Pente + Densite + Orient
##
```

```
##             Df Sum of Sq    RSS     AIC
## <none>                   7.4363 -37.174
## + Diametre  1  0.047495 7.3888 -35.386
## + Melange   1  0.024528 7.4118 -35.283
## + HautMax   1  0.011249 7.4251 -35.224
## + Hauteur   1  0.004636 7.4317 -35.195
## + NbPins    1  0.002154 7.4341 -35.184
```

```
summary(select.variables)
```

```
##
## Call:
## lm(formula = NbNids ~ NbStrat + Altitude + Pente + Densite +
##     Orient, data = chenilles)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.02011 -0.30225  0.09281  0.30246  1.45748
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.8986047  1.3963713   5.657 5.25e-06 ***
## NbStrat     -1.2869640  0.4148109  -3.103  0.00446 **
## Altitude    -0.0026117  0.0008919  -2.928  0.00685 **
## Pente       -0.0347266  0.0136941  -2.536  0.01731 *
## Densite      0.6608256  0.3539922   1.867  0.07283 .
## Orient      -0.7703655  0.5285038  -1.458  0.15648
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5248 on 27 degrees of freedom
## Multiple R-squared:  0.6425, Adjusted R-squared:  0.5763
## F-statistic: 9.704 on 5 and 27 DF,  p-value: 2.164e-05
```

Dans ce cas, on obtient comme modèle final un modèle où on retient comme variables : `NbStrat`, `Altitude`, `Pente`, `Densite` et `Orient`.

4.2 Compléments pour le modèle d'ANOVA à 1 facteur : tests de comparaisons multiples

On reprend l'exemple de l'influence du genre d'un individu sur sa taille introduit à la section 3.2.2.2 pour lequel on a proposé un modèle d'ANOVA à 1 facteur :

$$Y_{i,k} = \mu + \alpha_i + E_{i,k},$$

où

- $Y_{i,k}$ est la variable aléatoire qui modélise la taille du k-ième individu ayant le genre i,

- μ décrit l'effet moyen et α_i représente l'effet additif dû à la modalité i du facteur, et
- les $E_{i,k}$ sont i.i.d. de loi $\mathcal{N}(0, \sigma^2)$,

pour $1 \leq i \leq I = 2$ et $1 \leq k \leq n_i$ (n_i représente le nombre d'individus ayant le genre i : $n_1 = 29$ et $n_2 = 38$, en notant "1" le genre "femme" et "2" le genre "homme").

Dans ce modèle, il y a $I + 1$ paramètres : μ, α_1, $\alpha_2, \ldots, \alpha_I$. Dans R, la contrainte usuelle est : $\alpha_1 = 0$. Avec cette contrainte, $\widehat{\boldsymbol{\theta}}$ peut être calculé et le nombre de paramètres "libres" du modèles *i.e.* à estimer est $p = I$ (voir Section 3.3).

Lorque l'on fait le test du modèle, on souhaite tester :

$$(H_0) : "\alpha_1 = \alpha_2 = \cdots = \alpha_I" \text{ contre } (H_1) : "\exists i \neq j,\ \alpha_i \neq \alpha_j". \tag{4.1}$$

Dans le cas $I = 2$ de l'exemple sur l'influence du genre sur la taille, cela revient juste à tester :

$$(H_0) : "\alpha_1 = \alpha_2" \text{ contre } (H_1) : "\alpha_1 \neq \alpha_2".$$

Dans le cas $I = 3$, on peut effectuer les trois tests ci-dessous (d'où le nom de tests multiples) à la place de (4.1) :

$$\begin{aligned}
(H_0^1) &: "\alpha_1 = \alpha_2" \text{ contre } (H_1^1) : "\alpha_1 \neq \alpha_2" \\
(H_0^2) &: "\alpha_1 = \alpha_3" \text{ contre } (H_1^2) : "\alpha_1 \neq \alpha_3" \\
(H_0^3) &: "\alpha_2 = \alpha_3" \text{ contre } (H_1^3) : "\alpha_2 \neq \alpha_3".
\end{aligned}$$

Dans le cas général, il faut faire $I(I-1)/2$ tests.

Pour s'assurer que le test global (4.1) est de niveau α, chacun des "sous-tests" doit être effectué à un niveau plus faible que α noté α_ℓ comme on va le montrer ci-dessous.

$$\begin{aligned}
\mathbb{P}_{H_0}(\text{Rejeter } (H_0)) &= \mathbb{P}_{H_0}(\exists \ell,\ (H_0^\ell) \text{ soit rejetée}) = \mathbb{P}_{H_0}(\cup_\ell \{(H_0^\ell) \text{ soit rejetée}\}) \\
&\leq \sum_{\ell=1}^{L} \mathbb{P}_{H_0}(\{(H_0^\ell) \text{ soit rejetée}\}) = \sum_{\ell=1}^{L} \mathbb{P}_{H_0^\ell}(\{(H_0^\ell) \text{ soit rejetée}\}) = \sum_{\ell=1}^{L} \alpha_\ell,
\end{aligned}$$

où L est le nombre de tests à faire.

Si, par exemple, $\alpha_\ell = \alpha/L = 2\alpha/(I(I-1))$ alors on garantit que le test global (4.1) est de niveau α. Cette façon de choisir α_ℓ s'appelle la *correction de Bonferroni*.

```
pairwise.t.test(PT$taille,PT$genre,p.adjust.method ="bonferroni")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  PT$taille and PT$genre
##
##       femme
## homme 3.5e-15
##
## P value adjustment method: bonferroni
```

Dans le cas de l'exemple, le test correspond exactement au test du modèle et on conlut qu'il y a bien une différence significative entre les deux modalités de la variable genre sur la taille.

Dans les cas où le nombre de modalités I est supérieur ou égal à 3, les tests de comparaison multiples permettent de comparer 2 à 2 les effets des différentes modalités du facteur et sont donc plus informatifs que le test du modèle.

4.3 Modèle d'ANOVA à 2 facteurs

On utilise cette modélisation lorsque l'on souhaite étudier l'influence de 2 variables qualitatives ayant respectivement I et J modalités sur une variable quantitative Y. C'est une extension du modèle d'ANOVA à 1 facteur.

4.3.1 Introduction : exemple des grains de colza

On souhaite étudier l'effet du niveau de la fertilisation et de la rotation sur le poids des grains de colza. Les deux variables qualitatives sont

- le 1er facteur : la fertilisation ayant $I = 2$ modalités notées 1 pour faible et 2 pour forte
- le 2ème facteur : la rotation ayant $J = 3$ modalités notées A, B et C (A signifie sans enfouissement de paille ($j = 1$), B signifie avec enfouissement de paille ($j = 2$) et C signifie avec 4 années de prairie entre chaque succession sans enfouissement de paille ($j = 3$)).

Dans la suite, on note :

- $n_{i,j}$ le nombre d'observations pour la modalité i du 1er facteur et la modalité j du 2ème facteur,
- $n_{i+} = \sum_{j=1}^{J} n_{i,j}$ et $n_{+j} = \sum_{i=1}^{I} n_{i,j}$,
- $n = \sum_{i=1}^{I} \sum_{j=1}^{J} n_{ij}$ le nombre total d'observations.

On charge les données dans R comme suit.

```
colza<-read.table('Colza.txt',header=T)
head(colza)
```

```
##   Fertilisation Rotation PdsGrains
## 1             1        A      27.6
## 2             1        A      16.3
## 3             1        A      11.4
## 4             1        A      38.2
## 5             1        A      38.1
## 6             1        A      24.7
```

```
class(colza$Rotation)                  # On vérifie les types des variables
```

```
## [1] "character"
```

```
class(colza$PdsGrains)
```

```
## [1] "numeric"
```

```
class(colza$Fertilisation)
```

```
## [1] "integer"
```

```
# Pour changer la variable Fertilisation en variable qualitative
colza$Fertilisa<-as.factor(colza$Fertilisation)
class(colza$Fertilisa)
```

```
## [1] "factor"
```

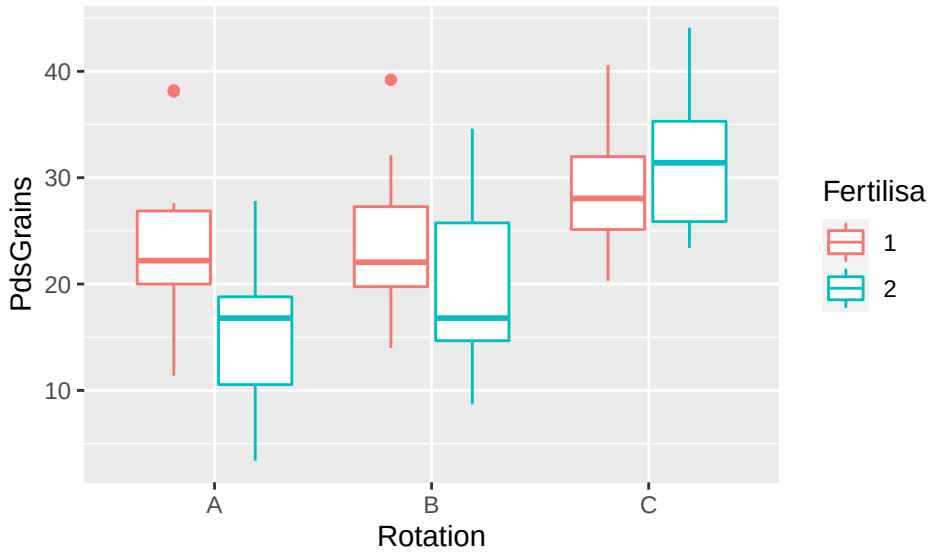
```
# Pour obtenir les effectifs aux différents croisements des modalités
# des variables qualitatives
table(colza$Rotation,colza$Fertilisa)
```

```
##
##      1  2
##   A 10 10
##   B 10 10
##   C 10 10
```

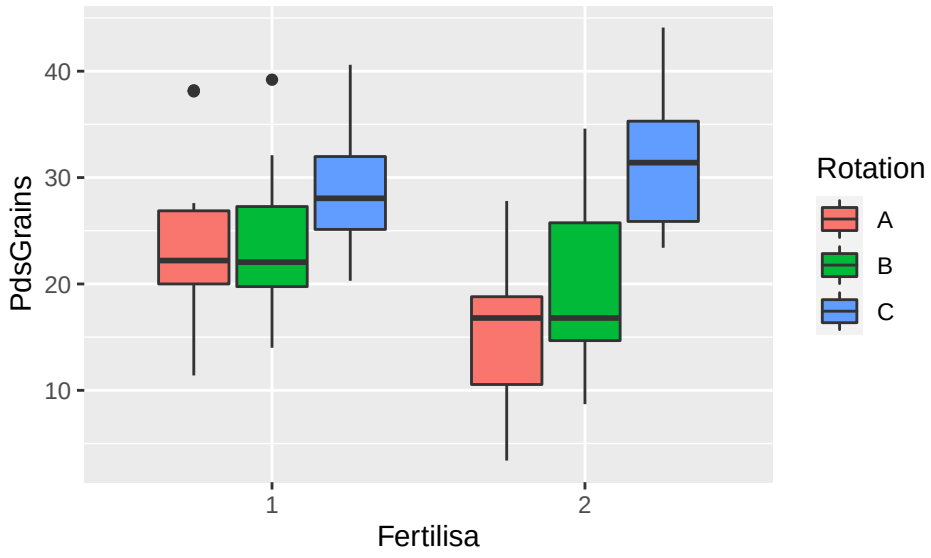
On observe ici que l'on a affaire à un **dispositif équilibré**. En effet, $n_{i,j}$ est égal à une constante pour tout croisement (i, j) qui vaut 10 dans cet exemple.

On peut représenter graphiquement les données de l'exemple.

```
ggplot(colza,aes(x=Rotation,y=PdsGrains,colour=Fertilisa))+geom_boxplot()
```




```
ggplot(colza,aes(x=Fertilisa,y=PdsGrains,fill=Rotation))+geom_boxplot()
```



4.3.2 Modélisation

4.3.2.1 Ecriture du modèle

En notant $Y_{i,j,k}$ la variable aléatoire qui représente le poids des k-ièmes grains de colza ayant reçu la fertilisation i et la rotation j, le modèle d'ANOVA à 2 facteurs s'écrit :

$$Y_{i,j,k} = \mu + \alpha_i + \beta_j + \gamma_{i,j} + E_{i,j,k},\ 1 \leq i \leq I,\ 1 \leq j \leq J,\ 1 \leq k \leq n_{i,j},$$

où les $E_{i,j,k}$ sont i.i.d. de loi $\mathcal{N}(0, \sigma^2)$. Ici,

- μ est l'effet moyen,
- α_i représente l'effet principal du 1er facteur (la fertilisation),
- β_j représente l'effet principal du 2ème facteur (la rotation),
- $\gamma_{i,j}$ représente l'interaction entre les deux facteurs.

Il s'agit bien d'un modèle linéaire comme on va le voir grâce à l'écriture matricielle ci-après.

4.3.2.2 Ecriture matricielle

Comme tout modèle linéaire, on peut écrire :

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\theta} + \mathbf{E},$$

où

$$\mathbf{Y} = \begin{pmatrix} Y_{111} \\ \vdots \\ Y_{11n_{11}} \\ Y_{121} \\ \vdots \\ Y_{12n_{12}} \\ \vdots \\ Y_{211} \\ \vdots \\ Y_{21n_{21}} \\ \vdots \\ Y_{231} \\ \vdots \\ Y_{23n_{23}} \end{pmatrix}, \ \mathbf{E} = \begin{pmatrix} E_{111} \\ \vdots \\ E_{11n_{11}} \\ E_{121} \\ \vdots \\ E_{12n_{12}} \\ \vdots \\ E_{211} \\ \vdots \\ E_{21n_{21}} \\ \vdots \\ E_{231} \\ \vdots \\ E_{23n_{23}} \end{pmatrix}, \ \boldsymbol{\theta} = \begin{pmatrix} \mu \\ \alpha_1 \\ \alpha_2 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \gamma_{11} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{21} \\ \gamma_{22} \\ \gamma_{23} \end{pmatrix}$$

et

$$\mathbf{X} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Les vecteurs $\mathbf{Y}$ et $\mathbf{E}$ sont de taille $n \times 1$, le vecteur $\boldsymbol{\beta}$ est de taille $(1 + I + J + IJ) \times 1$ et la matrice $\mathbf{X}$ est de taille $n \times (1 + I + J + IJ)$.

On observe que le rang de $\mathbf{X}$ vaut IJ : avec les IJ dernières colonnes on peut reconstituer toutes les colonnes précédentes. $\mathbf{X}$ n'est donc pas de rang plein, on va devoir mettre des contraintes sur les paramètres pour être en mesure de les estimer.

4.3.2.3 Contraintes

Les contraintes qui sont utilisées par défaut dans le logiciel R sont les suivantes :

$$\alpha_1 = 0,\ \beta_1 = 0,\ \forall j,\ \gamma_{1,j} = 0,\ \text{ et }\ \forall i,\ \gamma_{i,1} = 0,$$

qui correspondent respectivement à 1, 1, J et $(I-1)$ contrainte(s). En effet, la contrainte $\gamma_{1,1} = 0$ est écrite deux fois. Ainsi, on a bien : $I + J + 1$ contraintes.

4.3.3 Tests des effets

On peut proposer des tests de type I ou des tests de type II pour tester les différents effets.

- Tests de type I :

Paramètres	Df	Sum of Sq	Mean Sq	F-statistic
(α)	$I-1$	$R(\alpha\|\mu)$	$R(\alpha\|\mu)/(I-1)$	$\widehat{\sigma}^{-2}R(\alpha\|\mu)/(I-1)$
(β)	$J-1$	$R(\beta\|\mu,\alpha)$	$R(\beta\|\mu,\alpha)/(J-1)$	$\widehat{\sigma}^{-2}R(\beta\|\mu,\alpha)/(J-1)$
(γ)	$(I-1)(J-1)$	$R(\gamma\|\mu,\alpha,\beta)$	$R(\gamma\|\mu,\alpha,\beta)/((I-1)(J-1))$	$\widehat{\sigma}^{-2}R(\gamma\|\mu,\alpha,\beta)/((I-1)(J-1))$

- Tests de type II :

Paramètres	Df	Sum of Sq	Mean Sq	F-statistic
(α)	$I-1$	$R(\alpha\|\mu,\beta)$	$R(\alpha\|\mu,\beta)/(I-1)$	$\widehat{\sigma}^{-2}R(\alpha\|\mu,\beta)/(I-1)$
(β)	$J-1$	$R(\beta\|\mu,\alpha)$	$R(\beta\|\mu,\alpha)/(J-1)$	$\widehat{\sigma}^{-2}R(\beta\|\mu,\alpha)/(J-1)$
(γ)	$(I-1)(J-1)$	$R(\gamma\|\mu,\alpha,\beta)$	$R(\gamma\|\mu,\alpha,\beta)/((I-1)(J-1))$	$\widehat{\sigma}^{-2}R(\gamma\|\mu,\alpha,\beta)/((I-1)(J-1))$

Dans ces tables, on a :

$$\begin{aligned}
R(\alpha|\mu) &= SCR(\mu) - SCR(\mu,\alpha),\\
R(\beta|\mu,\alpha) &= SCR(\mu,\alpha) - SCR(\mu,\alpha,\beta),\\
R(\gamma|\mu,\alpha,\beta) &= SCR(\mu,\alpha,\beta) - SCR(\mu,\alpha,\beta,\gamma),\\
R(\alpha|\mu,\beta) &= SCR(\mu,\beta) - SCR(\mu,\alpha,\beta),
\end{aligned}$$

où, par exemple, la quantité $R(\alpha|\mu)$ désigne de combien diminue le SCR lorsque l'on ajoute le 1er facteur (α) au modèle nul (μ). De plus, "Df" signifie "Degrees of Freedom (degrés de liberté)", "Sum of Sq" désigne "Sum of Squares (sommes de carrés)", "Mean Sq" désigne "Mean squares (Moyenne des carrés)" et "F-statistic" correspond à la "statistique de Fisher" qui permet de tester l'effet du facteur de la ligne associée.

On rappelle que $\widehat{\sigma}^2 = SCR/(n-IJ)$, puisqu'il y a IJ paramètres libres dans ce modèle.

Dans le cas des tests de type I, contrairement aux tests de type II, on ajoute les effets au fur et à mesure.

Lorsque le dispositif est **orthogonal** *i.e.* lorsque $n_{ij} = \frac{n_{i+}n_{+j}}{n}$, pour tout (i,j), les tests de type I et II sont équivalents. Le cas équilibré est un cas particulier de dispositif orthogonal.

Lorsque l'on teste les effets, on teste d'abord le terme d'interaction.

4.3.4 Mise en oeuvre dans R

Il est important de vérifier que les variables explicatives sont bien considérées comme des variables qualitatives (avec le statut `factor` dans le logiciel R).

```
modA2I<-lm(PdsGrains ~ Fertilisa * Rotation, data=colza)
summary(modA2I)
```

```
##
## Call:
## lm(formula = PdsGrains ~ Fertilisa * Rotation, data = colza)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.710  -5.492  -1.125   3.752  15.200
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             24.110      2.377  10.141 4.16e-14 ***
## Fertilisa2              -8.300      3.362  -2.469   0.0168 *
## RotationB               -0.110      3.362  -0.033   0.9740
## RotationC                4.530      3.362   1.347   0.1835
## Fertilisa2:RotationB     4.140      4.755   0.871   0.3878
## Fertilisa2:RotationC    11.410      4.755   2.400   0.0199 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.518 on 54 degrees of freedom
## Multiple R-squared:  0.3522, Adjusted R-squared:  0.2923
## F-statistic: 5.873 on 5 and 54 DF,  p-value: 0.0002106
```

```
#analyse des effets
#type I
anova(modA2I)
```

```
## Analysis of Variance Table
##
## Response: PdsGrains
##                     Df  Sum Sq Mean Sq F value    Pr(>F)
## Fertilisa            1  145.70  145.70  2.5776 0.1142193
## Rotation             2 1180.48  590.24 10.4417 0.0001466 ***
## Fertilisa:Rotation   2  333.63  166.82  2.9511 0.0607686 .
## Residuals           54 3052.47   56.53
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#type II
library(car) #package a charger
```

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

```
Anova(modA2I)
```

```
## Anova Table (Type II tests)
##
## Response: PdsGrains
##                      Sum Sq Df F value    Pr(>F)
## Fertilisa            145.70  1  2.5776 0.1142193
## Rotation            1180.48  2 10.4417 0.0001466 ***
## Fertilisa:Rotation   333.63  2  2.9511 0.0607686 .
## Residuals           3052.47 54
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Au niveau 5% on enlève le terme d'interaction.

```
modA2sI<-lm(PdsGrains ~ Fertilisa + Rotation, data=colza)
summary(modA2sI)
```

```
##
## Call:
## lm(formula = PdsGrains ~ Fertilisa + Rotation, data = colza)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.002  -5.074  -1.428   5.287  16.682
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   21.518      2.008  10.718 3.49e-15 ***
## Fertilisa2    -3.117      2.008  -1.552  0.12622
## RotationB      1.960      2.459   0.797  0.42877
## RotationC     10.235      2.459   4.162  0.00011 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.776 on 56 degrees of freedom
## Multiple R-squared:  0.2814, Adjusted R-squared:  0.2429
## F-statistic: 7.311 on 3 and 56 DF,  p-value: 0.0003203
```

```
#type I
anova(modA2sI)
```

```
## Analysis of Variance Table
##
## Response: PdsGrains
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Fertilisa  1  145.7  145.70  2.4097 0.1262207
## Rotation   2 1180.5  590.24  9.7615 0.0002307 ***
## Residuals 56 3386.1   60.47
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#type II
Anova(modA2sI)
```

```
## Anova Table (Type II tests)
##
## Response: PdsGrains
##           Sum Sq Df F value    Pr(>F)
## Fertilisa  145.7  1  2.4097 0.1262207
## Rotation  1180.5  2  9.7615 0.0002307 ***
## Residuals 3386.1 56
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

On ne retient que le modèle avec la rotation.

```
modA1<-lm(PdsGrains ~ Rotation, data=colza)
summary(modA1)
```

```
##
## Call:
## lm(formula = PdsGrains ~ Rotation, data = colza)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.560  -5.314  -1.040   4.850  18.240
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   19.960      1.760  11.340 3.07e-16 ***
## RotationB      1.960      2.489   0.787 0.434310
## RotationC     10.235      2.489   4.112 0.000127 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.872 on 57 degrees of freedom
## Multiple R-squared:  0.2505, Adjusted R-squared:  0.2242
## F-statistic: 9.526 on 2 and 57 DF,  p-value: 0.0002697
```

On peut faire des tests de comparaison 2 à 2 des différentes modalités de la variable Rotation.

```
library(lsmeans)
lsmeans(modA1,pairwise~Rotation,adjust="bonferroni")
```

```
## $lsmeans
##  Rotation lsmean   SE df lower.CL upper.CL
##  A          20.0 1.76 57     16.4     23.5
##  B          21.9 1.76 57     18.4     25.4
```

```
##  C          30.2 1.76 57     26.7     33.7
##
## Confidence level used: 0.95
##
## $contrasts
##  contrast estimate   SE df t.ratio p.value
##  A - B       -1.96 2.49 57 -0.787  1.0000
##  A - C      -10.23 2.49 57 -4.112  0.0004
##  B - C       -8.28 2.49 57 -3.324  0.0047
##
## P value adjustment: bonferroni method for 3 tests
```

```
### Autre possibilité
# library(emmeans)
# emmeans(modA1,pairwise~Rotation,adjust='bonferroni')
```

4.4 Modèle d'ANCOVA

On utilise ce modèle lorsque l'on souhaite étudier l'influence de variables quantitatives et de variables qualitatives sur une variable quantitative.

4.4.1 Introduction : relation entre la taille et le poids en fonction du genre

On cherche à modéliser la relation existant entre le poids et la taille en tenant compte de la variable genre.

```
PT <- read.table("PT.txt", header=TRUE, sep="\t", dec=",")
head(PT)
```

```
##   genre   taille    poids
## 1 femme 61.11732 138.5426
## 2 femme 62.10180 127.9359
## 3 femme 62.11668 119.4516
## 4 femme 62.13176 109.8725
## 5 femme 61.80914 107.1128
## 6 femme 62.09234 104.1216
```

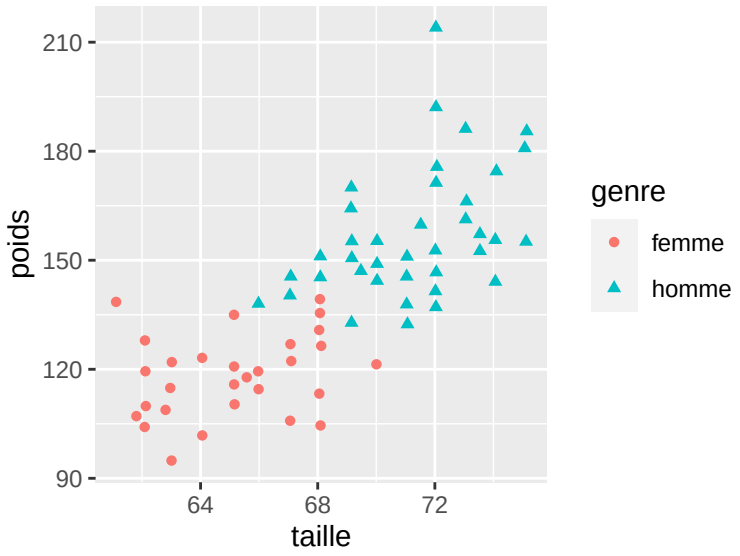
```
dim(PT)
```

```
## [1] 67  3
```

```
table(PT$genre)
```

```
##
## femme homme
##    29    38
```

```
ggplot(PT,aes(x=taille,y=poids,color=genre,shape=genre))+geom_point()
```



4.4.2 Modélisation

4.4.2.1 Ecriture du modèle

On cherche à expliquer une variable quantitative Y (dans notre exemple le poids) en fonction d'une variable quantitative x (dans notre exemple la taille) et d'une variable qualitative ayant I modalités (dans notre exemple le genre qui a 2 modalités : femme et homme). Le modèle d'ANCOVA s'écrit :

$$Y_{ik} = a_i + b_i x_{ik} + E_{ik},\ 1 \leq i \leq I,\ 1 \leq k \leq n_i, \tag{4.2}$$

où

- Y_{ik} est la variable aléatoire modélisant le poids du k-ième individu ayant le genre i,
- x_{ik} correspond à la taille du k-ième individu ayant le genre i,
- les E_{ik} sont des variables aléatoires i.i.d. de loi $\mathcal{N}(0, \sigma^2)$.

Dans ce modèle, il y a $2I$ paramètres. On peut aussi, comme pour les modèles d'analyse de la variance, proposer une version non identifiable du modèle (mais qui permet de dissocier les effets des différentes variables) en posant :

$$a_i = \mu + \alpha_i \quad \text{et} \quad b_i = \beta + \gamma_i.$$

On en déduit l'écriture suivante du modèle :

$$Y_{ik} = (\mu + \alpha_i) + (\beta + \gamma_i) x_{ik} + E_{ik}, \tag{4.3}$$

où il y a $1 + I + 1 + I = 2(I + 1)$ paramètres à estimer.

Il s'agit bien dans les deux cas d'un modèle linéaire comme on va le voir avec l'écriture matricielle ci-dessous.

4.4.2.2 Ecriture matricielle

On a :

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\theta} + \mathbf{E},$$

où

$$\mathbf{Y} = \begin{pmatrix} Y_{11} \\ \vdots \\ Y_{1n_1} \\ Y_{21} \\ \vdots \\ Y_{2n_2} \end{pmatrix}, \ \mathbf{X} = \begin{pmatrix} 1 & 1 & 0 & x_{11} & x_{11} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & x_{1n_1} & x_{1n_1} & 0 \\ 1 & 0 & 1 & x_{21} & 0 & x_{21} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & x_{2n_2} & 0 & x_{2n_2} \end{pmatrix} \text{ et } \boldsymbol{\theta} = \begin{pmatrix} \mu \\ \alpha_1 \\ \alpha_2 \\ \beta \\ \gamma_1 \\ \gamma_2 \end{pmatrix},$$

dans le cas où $I = 2$.

Comme pour les modèles d'ANOVA, on constate que la matrice $\mathbf{X}$ n'est pas de rang plein : il n'y a que $2I$ colonnes indépendantes parmi les $2(I+1)$. Il nous faut donc 2 contraintes. Dans R, les contraintes usuelles sont :

$$\alpha_1 = 0 \ \text{ et } \ \gamma_1 = 0.$$

4.4.3 Tests des différents effets

On utilise exactement la même démarche que celle mise en place dans l'étude d'un modèle d'ANOVA à deux facteurs.

4.4.3.1 Test de l'interaction (γ)

On teste

$$(H_0) : \text{"}\forall i, \ \gamma_i = 0\text{" contre } (H_1) : \text{"}\exists i, \ \gamma_i \neq 0\text{"}.$$

On utilise pour cela la statistique de Fisher suivante :

$$F = \frac{R(\gamma|\mu,\alpha,\beta)/(I-1)}{\widehat{\sigma}^2} \overset{(H_0)}{\hookrightarrow} \mathcal{F}(I-1, n-2I).$$

Si (H_0) n'est pas rejetée, on peut tester les autres effets.

4.4.3.2 Test de l'effet de x (β)

On teste

$$(H_0) : \text{"}\beta = 0\text{" contre } (H_1) : \text{"}\beta \neq 0\text{"}.$$

On utilise pour cela la statistique de Fisher suivante :

$$F = \frac{R(\beta|\mu,\alpha)/1}{\widehat{\sigma}^2_{-\gamma}} \overset{(H_0)}{\hookrightarrow} \mathcal{F}(1, n-I-1).$$

où $\widehat{\sigma}^2_{-\gamma}$ désigne l'estimateur de σ^2 dans le modèle d'ANCOVA sans interaction.

4.4.3.3 Test de l'effet du facteur (α)

On teste

$$(H_0) : \text{"}\forall i,\ \alpha_i = 0\text{" contre } (H_1) : \text{"}\exists i,\ \alpha_i \neq 0\text{".}$$

En type II, on utilise par exemple la statistique de Fisher suivante :

$$F = \frac{R(\alpha|\mu,\beta)/(I-1)}{\widehat{\sigma}^2_{-\gamma}} \overset{(H_0)}{\hookrightarrow} \mathcal{F}(I-1, n-I-1).$$

Remarque 7. *Avant d'effectuer ces tests, en type I ou en type II, on vérifie le sens qu'on leur donne.*

4.4.4 Comparaison des groupes avec les moyennes brutes et ajustées

Dans les cas où le nombre de modalités I du facteur est supérieur ou égal à 3, on souhaite comparer les moyennes des groupes 2 à 2.

Dans le modèle d'analyse de la covariance, la moyenne du groupe i (dans l'exemple, la moyenne des poids des individus de genre i) est :

$$\mu_{i\bullet} = \frac{1}{n_i}\sum_{k=1}^{n_i} \mathbb{E}(Y_{ik}) = \mu + \alpha_i + (\beta + \gamma_i)x_{i\bullet},$$

avec $x_{i\bullet} = \dfrac{1}{n_i}\displaystyle\sum_{k=1}^{n_i} x_{ik}$ la valeur moyenne de la variable x pour le groupe i.

Lorsque l'on compare deux groupes en testant "$\mu_{i\bullet} = \mu_{i'\bullet}$", on s'intéresse à l'écart entre ces deux groupes en prenant en compte l'écart en abscisse $(x_{i\bullet} - x_{i'\bullet})$. Si cet écart en abscisse est dû au hasard de l'échantillonnage, il n'est pas pertinent d'utiliser ces moyennes (dites brutes ou classiques) pour comparer les groupes.

On propose alors de comparer les moyennes ajustées définies par

$$\tilde{\mu}_{i\bullet} = \mu + \alpha_i + (\beta + \gamma_i)x_{\bullet\bullet},$$

avec $x_{\bullet\bullet} = \dfrac{1}{n}\displaystyle\sum_{i=1}^{I}\sum_{k=1}^{n_i} x_{ik}$ la valeur moyenne de la variable x pour toutes les observations. Ainsi, on compare l'écart de réponses entre les groupes obtenues pour une même abscisse x (la moyenne générale).

4.4.5 Mise en oeuvre dans R

```
poidstaille.lm=lm(poids~genre+taille+genre:taille,data=PT)
summary(poidstaille.lm)
```

```
##
## Call:
## lm(formula = poids ~ genre + taille + genre:taille, data = PT)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.998  -8.544  -1.154   6.202  54.412
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         41.088     71.077   0.578    0.565
## genrehomme        -121.300    100.687  -1.205    0.233
## taille               1.186      1.090   1.088    0.281
## genrehomme:taille    2.143      1.480   1.448    0.153
##
## Residual standard error: 14.38 on 63 degrees of freedom
## Multiple R-squared:  0.6753, Adjusted R-squared:  0.6598
## F-statistic: 43.68 on 3 and 63 DF,  p-value: 2.167e-15
```

```
### Tests de type II
library(car)
Anova(poidstaille.lm)
```

```
## Anova Table (Type II tests)
##
## Response: poids
##               Sum Sq Df F value    Pr(>F)
## genre         3713.2  1 17.9628 7.495e-05 ***
## taille        2100.5  1 10.1613  0.002234 **
## genre:taille   433.6  1  2.0975  0.152501
## Residuals    13023.3 63
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

On peut donc enlever le terme d'interaction et on recommence l'analyse.

```
# Modèle sans interaction
poidstaille.sansint.lm=lm(poids~genre+taille,data=PT)
summary(poidstaille.sansint.lm)
```

```
##
## Call:
## lm(formula = poids ~ genre + taille, data = PT)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.176  -8.233  -1.606   7.770  55.183
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -34.7090    48.5036  -0.716   0.4768
## genrehomme   24.2858     5.7791   4.202 8.35e-05 ***
## taille        2.3498     0.7434   3.161   0.0024 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.5 on 64 degrees of freedom
## Multiple R-squared:  0.6645, Adjusted R-squared:  0.654
## F-statistic: 63.38 on 2 and 64 DF,  p-value: 6.64e-16
```

```
# Tests de type II
Anova(poidstaille.sansint.lm)
```

```
## Anova Table (Type II tests)
##
## Response: poids
##            Sum Sq Df F value    Pr(>F)
## genre      3713.2  1   17.66 8.353e-05 ***
## taille     2100.5  1    9.99  0.002405 **
## Residuals 13456.8 64
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

On retient donc ce dernier modèle.

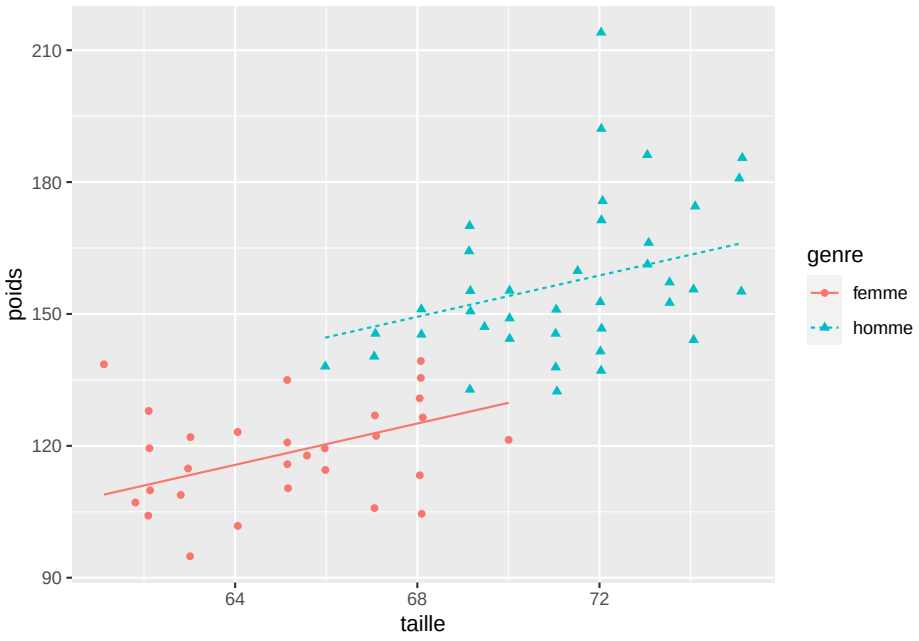
```
mean(PT$taille)
```

```
## [1] 68.60526
```

```
# Comparaison des groupes
library(emmeans)
emmeans(poidstaille.sansint.lm, pairwise ~ genre)
```

```
## $emmeans
##  genre emmean   SE df lower.CL upper.CL
##  femme    126 3.73 64      119      134
##  homme    151 3.07 64      145      157
##
## Confidence level used: 0.95
##
## $contrasts
##  contrast      estimate   SE df t.ratio p.value
##  femme - homme    -24.3 5.78 64 -4.202  0.0001
```

```
# Graphe d'ajustement
PT$predicted=predict(poidstaille.sansint.lm)
ggplot(PT, aes(x=taille, y=poids, shape=genre, color=genre))+geom_point()+
  geom_line(aes(x = taille, y = predicted, linetype=genre, color=genre))
```



4.4.6 Modèle d'ANCOVA avec plusieurs variables quantitatives

On aura parfois à considérer un modèle d'ANCOVA plus général *i.e.* avec 3 variables quantitatives (ici x, t et z). Dans ce cas, le modèle s'écrit comme suit :

$$Y_{i,k} = (\mu + \alpha_i) + (\beta + \gamma_i)x_{ik} + (\lambda + \eta_i)t_{ik} + (\nu + \theta_i)z_{ik} + E_{ik},$$

où les E_{ij} sont i.i.d. de loi $\mathcal{N}(0, \sigma^2)$.

Chapitre 5

Régression logistique

5.1 Introduction

Dans les modèles linéaires vus précédemment on a supposé que les observations pouvaient être modélisées par des variables aléatoires gaussiennes. Dans certaines applications, c'est un résultat binaire (ou dichotomique) d'une expérience que l'on souhaite mettre en relation avec des variables explicatives :

- des patients peuvent survivre ou décéder en fonction de différentes thérapies et de différents facteurs de risque (âge, le fait de fumer, sexe,...) qui peuvent ainsi être considérés comme des variables qui contribuent à expliquer la survie ou le décès des patients.
- des personnes peuvent avoir ou ne pas avoir d'emploi selon leur âge, sexe, type de formation,...
- un appareil peut fonctionner ou ne pas fonctionner ; cet état peut être mis en relation avec sa date de mise en service, les conditions dans laquelle il a fonctionné, ...

Dans ce cas, les modèles gaussiens ne sont pas adaptés. C'est le but de la régression logistique de pouvoir modéliser ce type de situation.

Un premier exemple

On dispose de la superficie (variable : `area`) de 50 îles différentes et de leur distance au continent (variable : `isolation`) et on s'intéresse à la nidification d'une espèce d'oiseaux (variable : `incidence`) sur chacune d'elles. Cette variable vaut 1 si l'espèce d'oiseaux est présente et 0 si celle-ci est absente.

```
##### Lecture des donnees "isolation.txt"
island<-read.table("isolation.txt", header=TRUE)
head(island)
```

```
##   incidence  area isolation
## 1         1 7.928     3.317
## 2         0 1.925     7.554
## 3         1 2.045     5.883
## 4         0 4.781     5.932
```

```
## 5          0 1.536     5.308
## 6          1 7.369     4.934
```

On représente les données dans la figure 5.1.

```
#### Graphes avec ggplot2
library(ggplot2)
library(gridExtra)
plot1=ggplot(island, aes(x=isolation, y=incidence))+geom_point()
plot2=ggplot(island,aes(x=incidence, y=isolation, group=incidence))+geom_boxplot()
grid.arrange(plot1, plot2, nrow=1, ncol=2)
```

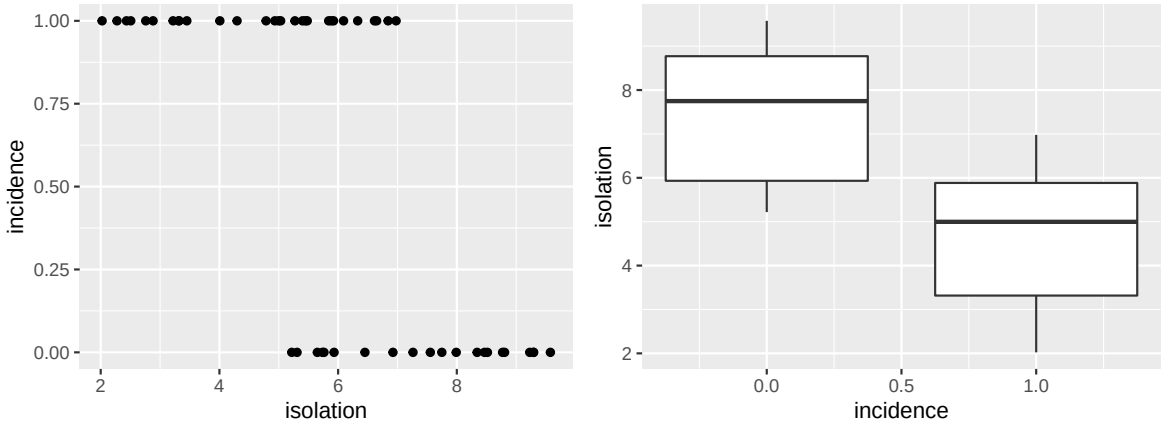


FIGURE 5.1 – Représentation des données avec ggplot.

Il serait maladroit d'appliquer un modèle de régression linéaire simple sur ces données comme le montre le graphe de gauche de la figure 5.2.

```
### Regression lineaire simple
reglin=lm(incidence~isolation,data=island)
summary(reglin)
```

```
##
## Call:
## lm(formula = incidence ~ isolation, data = island)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68589 -0.20650 -0.01367  0.31295  0.60621
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.55305    0.15631   9.936 3.12e-13 ***
## isolation   -0.16616    0.02519  -6.595 3.07e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.3649 on 48 degrees of freedom
## Multiple R-squared:  0.4754, Adjusted R-squared:  0.4645
## F-statistic: 43.49 on 1 and 48 DF,  p-value: 3.074e-08
```

```
par(mfrow=c(1,2))
x=2:10
droite=reglin$coefficient[1]+reglin$coefficient[2]*x
plot(x,droite,type='l',cex=3.0,xlab='isolation',ylab='incidence')
with(island,points(isolation,incidence,cex=1.0,pch=20,xlab='isolation',ylab='incidence'))
plot(x,courbe,type='l',cex=3.0,xlab='isolation',ylab='incidence')
with(island,points(isolation,incidence, cex=1.0,pch=20,xlab='isolation',ylab='incidence'))
```

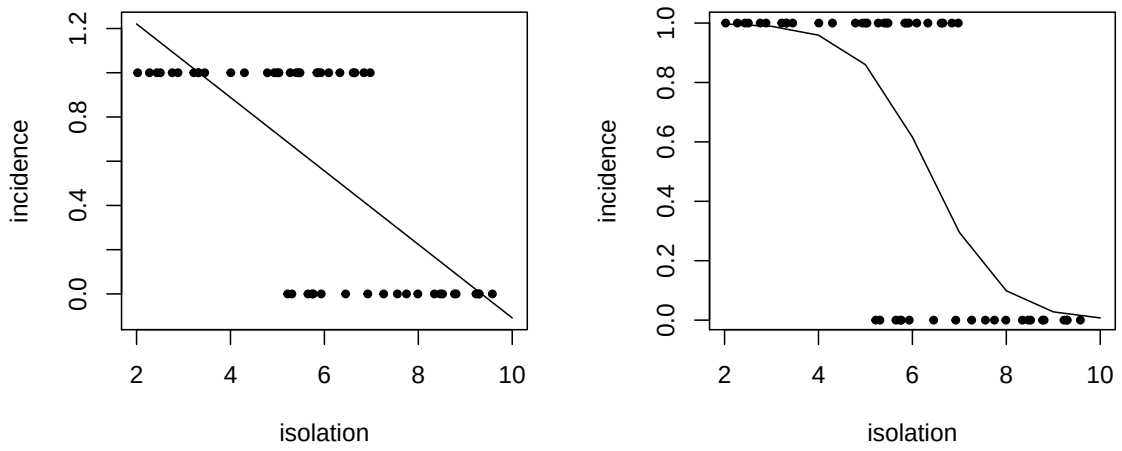


FIGURE 5.2 – Régression linéaire simple (gauche) et régression logistique (droite)

5.2 Modélisation

On modélise Y (v.a. à expliquer, ici l'incidence) par une v.a. de loi de Bernoulli de paramètre $\pi(x)$ notée Bernoulli($\pi(x)$) dans la suite où

$$\pi(x) = g(\beta_0 + \beta_1 x) \ ,$$

avec

$$g(x) = \frac{\exp(x)}{1+\exp(x)} = \frac{1}{1+\exp(-x)} \ ,$$

où g est appelée **fonction de lien**.

Ainsi,

$$\mathbb{P}(Y=1) = \frac{\exp(\beta_0+\beta_1 x)}{1+\exp(\beta_0+\beta_1 x)} \ ,$$

où Y est la variable `incidence` et x est la variable `isolation` dans l'exemple précédent.

Avec une telle modélisation,

$$\mathbb{E}(Y_i) = \mathbb{P}(Y_i = 1) = g(\beta_0 + \beta_1 x_i)$$

et la fonction de lien sert à garantir que $g(\beta_0 + \beta_1 x_i) \in [0, 1]$.

On modélise les n valeurs d'incidence par n va indépendantes de loi de Bernoulli

$$Y_i \overset{\text{indépendantes}}{\hookrightarrow} \text{Bernoulli}(g(\beta_0 + \beta_1 x_i)).$$

La terminologie "logistique" vient de l'anglais LOGIT (LOGarithm of Inverse Transformation) qui désigne la fonction inverse de g c'est-à-dire h définie par :

$$h(u) = \log\left(\frac{u}{1-u}\right).$$

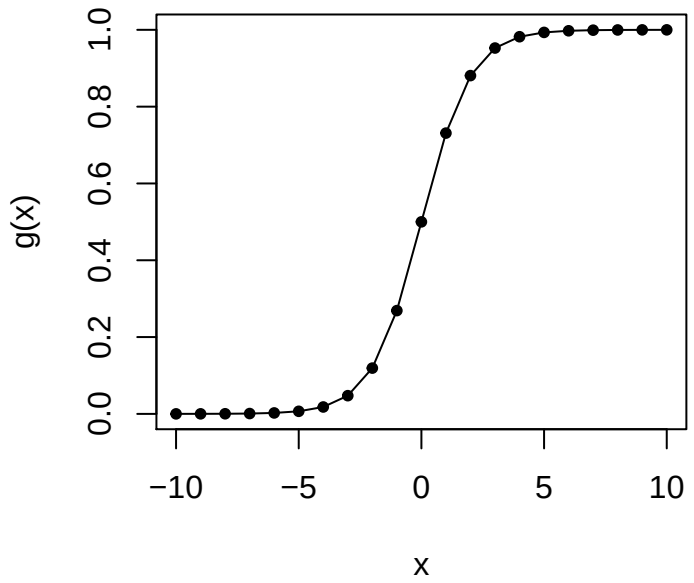


FIGURE 5.3 – Fonction g

5.3 Estimation des paramètres

On utilise la méthode du maximum de vraisemblance : soient Y_i des v.a. i.i.d de loi de Bernoulli($\pi_\beta(x_i)$) où

$$\pi_\beta(x_i) = \frac{\exp(\beta_0 + \beta_1 x_i)}{1 + \exp(\beta_0 + \beta_1 x_i)} = \frac{1}{1 + \exp\{-(\beta_0 + \beta_1 x_i)\}}.$$

La vraisemblance du modèle est donnée par :

$$L(Y_1,\dots,Y_n;\beta_0,\beta_1) \overset{\text{notation}}{=} L(Y_{1:n};\beta) = \prod_{i=1}^{n}\left\{\pi_\beta(x_i)^{Y_i}(1-\pi_\beta(x_i))^{1-Y_i}\right\} .$$

La log-vraisemblance est donnée par :

$$\begin{aligned}
\ell(Y_{1:n};\beta) &= \ell(Y_{1:n};\beta_0,\beta_1) \\
&= \sum_{i=1}^{n}\{Y_i\log(\pi_\beta(x_i)) + (1-Y_i)\log(1-\pi_\beta(x_i))\} \\
&= \sum_{i=1}^{n}\left\{Y_i\log\left(\frac{\pi_\beta(x_i)}{1-\pi_\beta(x_i)}\right) + \log(1-\pi_\beta(x_i))\right\}.
\end{aligned}$$

On observe que :

$$\pi_\beta(x_i) = \frac{1}{1+\exp\{-(\beta_0+\beta_1 x_i)\}} \text{ et } 1-\pi_\beta(x_i) = \frac{\exp\{-(\beta_0+\beta_1 x_i)}{1+\exp\{-(\beta_0+\beta_1 x_i)\}}$$

donc

$$\log\left(\frac{\pi_\beta(x_i)}{1-\pi_\beta(x_i)}\right) = \log\{\exp(\beta_0+\beta_1 x_i)\} = \beta_0+\beta_1 x_i .$$

Ainsi,

$$\begin{aligned}
\ell(Y_{1:n};\beta_0,\beta_1) &= \sum_{i=1}^{n}\left\{Y_i(\beta_0+\beta_1 x_i) + \log\left(\frac{1}{1+\exp(\beta_0+\beta_1 x_i)}\right)\right\} \\
&= \sum_{i=1}^{n}\{Y_i(\beta_0+\beta_1 x_i) - \log(1+\exp(\beta_0+\beta_1 x_i))\} .
\end{aligned}$$

On calcule le gradient de ℓ

$$\frac{\partial\ell(Y_{1:n};\beta_0,\beta_1)}{\partial\beta_0} = \sum_{i=1}^{n}\left\{Y_i - \frac{\exp(\beta_0+\beta_1 x_i)}{1+\exp(\beta_0+\beta_1 x_i)}\right\} = \sum_{i=1}^{n}(Y_i-\pi_\beta(x_i)),$$

$$\frac{\partial\ell(Y_{1:n};\beta_0,\beta_1)}{\partial\beta_1} = \sum_{i=1}^{n}\left\{Y_i x_i - x_i\frac{\exp(\beta_0+\beta_1 x_i)}{1+\exp(\beta_0+\beta_1 x_i)}\right\} = \sum_{i=1}^{n}(Y_i-\pi_\beta(x_i))x_i.$$

$\hat\beta_0$ et $\hat\beta_1$ sont donc solution du système

$$\begin{cases} \sum_{i=1}^{n}\left(Y_i - \frac{\exp(\hat\beta_0+\hat\beta_1 x_i)}{1+\exp(\hat\beta_0+\hat\beta_1 x_i)}\right) = 0, \\ \sum_{i=1}^{n} x_i\left(Y_i - \frac{\exp(\hat\beta_0+\hat\beta_1 x_i)}{1+\exp(\hat\beta_0+\hat\beta_1 x_i)}\right) = 0. \end{cases}$$

On résumera ce système d'équations dans la suite par :

$$\nabla \ell(\hat{\beta}) = 0 \ , \ \text{où} \ \hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1) \ .$$

Ces expressions sont non linéaires en $\hat{\beta}_0$ et $\hat{\beta}_1$ contrairement à la régression linéaire. On doit donc faire appel à des méthodes de résolution numérique itératives telles que **l'algorithme de Newton-Raphson** par exemple.

5.4 Algorithme de Newton-Raphson

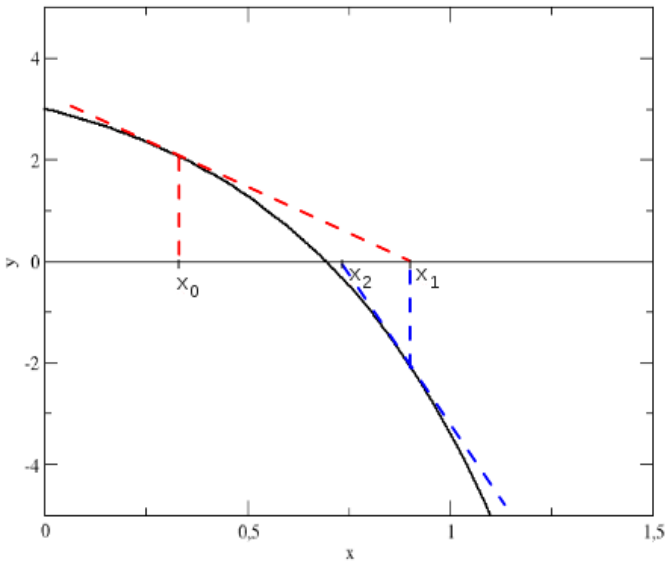
Problème : Trouver une racine d'une fonction $f : \mathbb{R} \to \mathbb{R}$ c'est-à-dire un point $x^\star$ tel que $f(x^\star) = 0$.

Idée : Approximer la fonction par sa tangente en utilisant un développement de Taylor au premier ordre

$$f(x) \approx f(x_0) + (x - x_0) f'(x_0) \Rightarrow 0 = f(x_0) + (x - x_0) f'(x_0) \ .$$

Algorithme itératif :

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$



Lorsque f est remplacée par f', l'algorithme de Newton-Raphson dans le cas scalaire s'écrit :

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} \ .$$

Dans le cas vectoriel, il devient :

$$\hat{\beta}^{(k+1)} = \hat{\beta}^{(k)} - \left[H(\hat{\beta}^{(k)}) \right]^{-1} \nabla \ell(\hat{\beta}^{(k)}) \ ,$$

où H (matrice hessienne) est définie par :

$$H(\beta) = \begin{pmatrix} \frac{\partial^2 \ell(Y_{1:n};\beta_0,\beta_1)}{\partial \beta_0^2} & \frac{\partial^2 \ell(Y_{1:n};\beta_0,\beta_1)}{\partial \beta_0 \partial \beta_1} \\ \frac{\partial^2 \ell(Y_{1:n};\beta_0,\beta_1)}{\partial \beta_0 \partial \beta_1} & \frac{\partial^2 \ell(Y_{1:n};\beta_0,\beta_1)}{\partial \beta_1^2} \end{pmatrix} .$$

Calcul de la matrice hessienne :

$$\begin{aligned} \frac{\partial \ell(Y_{1:n};\beta_0,\beta_1)}{\partial \beta_1} &= \sum_{i=1}^n x_i \left\{ Y_i - \frac{1}{1+\exp\{-(\beta_0+\beta_1 x_i)\}} \right\} \\ &= \sum_{i=1}^n x_i \left\{ Y_i - \varphi(\beta_0+\beta_1 x_i) \right\} . \end{aligned}$$

D'où

$$\frac{\partial^2 \ell(Y_{1:n};\beta_0,\beta_1)}{\partial \beta_1^2} = -\sum_{i=1}^n x_i^2 \varphi'(\beta_0+\beta_1 x_i) = -\sum_{i=1}^n x_i^2 \pi_\beta(x_i)(1-\pi_\beta(x_i) .$$

Ainsi

$$\begin{aligned} H(\beta) &= -\begin{pmatrix} \sum_{i=1}^n \pi_\beta(x_i)(1-\pi_\beta(x_i)) & \sum_{i=1}^n x_i \pi_\beta(x_i)(1-\pi_\beta(x_i)) \\ \sum_{i=1}^n x_i \pi_\beta(x_i)(1-\pi_\beta(x_i)) & \sum_{i=1}^n x_i^2 \pi_\beta(x_i)(1-\pi_\beta(x_i) \end{pmatrix} \\ &= -X^T W_\beta X , \end{aligned}$$

où $W_\beta = \text{diag}\{\pi_\beta(x_i)(1-\pi_\beta(x_i)\}_{1\leq i\leq n}$ et

$$X = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} .$$

5.4.1 Dans R

```
### Regression logistique
reglog=glm(incidence~isolation,data=island,family=binomial(link="logit"))
summary(reglog)
```

```
##
## Call:
## glm(formula = incidence ~ isolation, family = binomial(link = "logit"),
##     data = island)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8537  -0.3590   0.1168   0.6272   1.5476
```

```
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   8.5223     2.4511   3.477 0.000507 ***
## isolation    -1.3416     0.3926  -3.417 0.000633 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 68.029  on 49  degrees of freedom
## Residual deviance: 36.640  on 48  degrees of freedom
## AIC: 40.64
##
## Number of Fisher Scoring iterations: 6
```

On a :

$\widehat{\beta}_0$=8.5223 et $\widehat{\beta}_1$=-1.3416.

5.4.2 Déviance

La déviance d'un modèle $\mathcal{M}$ est définie par :

$$2(\ell_{\text{sat}} - \ell_{\mathcal{M}}),$$

où

$$\ell_{\mathcal{M}} = \sum_{i=1}^{n} \{Y_i \log(\pi_\beta(x_i)) + (1 - Y_i) \log(1 - \pi_\beta(x_i))\} \text{ et } \ell_{\text{sat}} = \sum_{i=1}^{n} \{Y_i \log(Y_i) + (1 - Y_i) \log(1 - Y_i)\} = 0 \text{ (ici).}$$

5.4.2.1 Déviance nulle

La déviance nulle est définie par :

$$2(\ell_{\text{sat}} - \ell_0),$$

où ℓ_0 est la log-vraisemblance sous l'hypothèse nulle :"$\beta_1 = 0$" :

$$\ell_0 = \sum_{i=1}^{n} \left\{ Y_i \log\left(\pi_{\widehat{\beta}_0}(x_i)\right) + (1 - Y_i) \log(1 - \pi_{\widehat{\beta}_0}(x_i)) \right\}.$$

On peut montrer que :

$$\widehat{\beta}_0 = \log\left(\frac{\bar{Y}}{1 - \bar{Y}}\right).$$

D'où :

$$\ell_0 = \sum_{i=1}^{n} \left\{ Y_i \widehat{\beta}_0 + \log\left(\frac{1}{1 + \exp(\widehat{\beta}_0)}\right) \right\} = \sum_{i=1}^{n} \left\{ Y_i \log\left(\frac{\bar{Y}}{1 - \bar{Y}}\right) + \log(1 - \bar{Y}) \right\}.$$

```
moy=mean(island$incidence)
log1=log(moy/(1-moy))
log2=log(1-moy)
l0=sum(island$incidence*log1)+length(island$incidence)*log2
lsat=0
null_dev=2*(lsat-l0)
null_dev
```

```
## [1] 68.0292
```

5.4.2.2 Déviance résiduelle

Elle est définie par :

$$2(\ell_{\text{sat}} - \ell_{\widehat{\beta}_0,\widehat{\beta}_1}).$$

où

$$\ell_{\widehat{\beta}_0,\widehat{\beta}_1} = \sum_{i=1}^{n} \left\{ Y_i(\widehat{\beta}_0 + \widehat{\beta}_1 x_i) - \log\left(1 + \exp(\widehat{\beta}_0 + \widehat{\beta}_1 x_i)\right) \right\}$$

```
beta0=reglog$coefficients[1]
beta1=reglog$coefficients[2]
lvrais1=sum(island$incidence*(beta0+beta1*island$isolation))
lvrais2=sum(log(1+exp(beta0+beta1*island$isolation)))
lmodele=lvrais1-lvrais2
dev_res=2*(lsat-lmodele)
dev_res
```

```
## [1] 36.63969
```

5.4.2.3 Critère d'Akaike (AIC)

Il est défini par :

$$AIC = 2(-\ell_{\mathcal{M}} + p),$$

où p est le nombre de paramètres du modèle $\mathcal{M}$. Dans notre exemple,

$$AIC = 2(-\ell_{\widehat{\beta}_0,\widehat{\beta}_1} + 2),$$

```
AIC=2*(-lmodele+2)
AIC
```

```
## [1] 40.63969
```

5.5 Test

On souhaite tester : (H_0) : "$\beta_1 = 0$" contre (H_1) : "$\beta_1 \neq 0$".

Pour cela on utilise le test du rapport de vraisemblance :

$$\Lambda_n = 2\left(\ell(Y_{1:n}; \hat{\beta}_0, \hat{\beta}_1) - \ell(Y_{1:n}; \tilde{\beta}_0, 0)\right)$$

où $\tilde{\beta}_0$ est l'estimateur de β_0 sous (H_0). On a donc :

$$\ell(Y_{1:n}; \hat{\beta}_0, \hat{\beta}_1) = \sum_{i=1}^{n} \left\{ Y_i \log(\pi_{\hat{\beta}}(x_i)) + (1 - Y_i) \log(1 - \pi_{\hat{\beta}}(x_i)) \right\}$$

et

$$\ell(Y_{1:n}; \tilde{\beta}_0, 0) = \sum_{i=1}^{n} \left\{ Y_i \log(\bar{Y}) + (1 - Y_i) \log(1 - \bar{Y}) \right\}.$$

On observe que

$$\Lambda_n = \text{déviance nulle} - \text{déviance résiduelle}.$$

Pour le mettre en oeuvre, on utilise que sous (H_0), Λ_n se comporte comme un $\chi^2(1)$ lorsque $n \to \infty$.

Dans R, pour notre exemple :

```
anova(reglog,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: incidence
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                         49     68.029
## isolation  1    31.39        48     36.640 2.111e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La p-valeur vaut 2.1111888×10^{-8} donc on rejette (H_0) d'où l'on déduit l'influence de l'isolation sur l'incidence.

Chapitre 6

Régression et classification par la méthode des K plus proches voisins (KNN)

6.1 Introduction

La méthode des K plus proches voisins est une méthode d'apprentissage supervisé notée en abrégé KNN, de l'anglais *K nearest neighbors*. Un algorithme d'apprentissage supervisé (par opposition à un algorithme d'apprentissage non supervisé) est un algorithme qui s'appuie sur des données d'entrée étiquetées pour **apprendre** une fonction qui produit une sortie appropriée lorsque de nouvelles données non étiquetées sont fournies.

L'algorithme des KNN peut aussi bien être utilisé pour la régression que pour la classification. Soit un échantillon de données $\mathcal{D} = (x_i, y_i)_{1 \leq i \leq n}$. Pour chaque entrée (ou prédicteur) $x_i \in \mathbb{R}^p$, on dispose d'une sortie y_i :

- si la sortie est de nature qualitative, c'est un problème de **classification** (on parle alors de classes, labels ou groupes) ;
- si la sortie est de nature quantitative, c'est un problème de **régression**.

Pour prédire la variable y associée à une nouvelle observation x (qui ne fait pas partie des données $\mathcal{D}$), l'algorithme des KNN consiste à prendre en compte les K éléments de $\mathcal{D}$ dont les entrées x_i sont les plus proches de la nouvelle observation x, **selon une distance à définir**. Par exemple, dans un problème de classification, on retiendra la classe la plus représentée parmi les K éléments associés aux K entrées les plus proches de la nouvelle observation x. On parle alors de **vote majoritaire**. Dans un problème de régression, c'est la **moyenne (ou la médiane)** des variables y_i des K plus proches entrées qui servira de prédiction. Ainsi, pour effectuer une prédiction, contrairement à la régression linéaire ou à la régression logistique, l'algorithme des KNN n'a pas besoin de construire de modèle prédictif.

L'algorithme des K plus proches voisins est simple et facile à mettre en œuvre mais est particulièrement sensible à la structure locale des données.

6.2 Algorithme des KNN

On peut résumer l'algorithme des K plus proches voisins comme suit.

Données en entrée :

— un ensemble de données $\mathcal{D}$,
— une distance d et
— un nombre entier K.

Algorithme des KNN :

Pour une nouvelle observation x dont on veut prédire sa variable de sortie y, on exécute les étapes suivantes :

1. Calculer toutes les distances de cette observation x avec les autres observations x_i du jeu de données $\mathcal{D}$.
2. Retenir les K observations du jeu de données $\mathcal{D}$ dont les x_i sont les plus proches de x selon la distance d.
3. Prendre les valeurs des y_i des K observations x_i retenues :
   — si on effectue une régression, calculer la moyenne (ou la médiane) des y_i retenues, ou
   — si on effectue une classification, effectuer un vote majoritaire parmi les y_i retenues.

En sortie :

On retourne la valeur calculée à l'étape 3 et il s'agit de la valeur prédite par l'algorithme des KNN pour l'observation x.

6.3 Choix de la distance

L'algorithme des KNN nécessite de calculer une distance entre deux observations : plus deux observations sont "proches" l'une de l'autre, plus elles sont similaires et auront possiblement la même sortie y (dans un problème de classification) ou des sorties "proches" (dans un problème de régression).

Il existe plusieurs distances : notamment, la plus usuelle, la distance euclidienne, mais aussi la distance de Manhattan, la distance de Minkowski, ... On choisit la fonction de distance en fonction du type des observations qu'on manipule. Ainsi pour des observations quantitatives, la distance euclidienne est un bon candidat et c'est celle qui est la plus utilisée.

Définition 16 (Distance euclidienne)**.** *La distance euclidienne entre deux observations* $x_i = (x_{i1}, \ldots, x_{ip})'$ *et* $x_{i'} = (x_{i'1}, \ldots, x_{i'p})'$ *de* $\mathbb{R}^p$ *est :*

$$d(x_i, x_{i'}) = \|x_i - x_{i'}\| = \sqrt{\sum_{j=1}^{p} (x_{ij} - x_{i'j})^2}.$$

Définition 17 (Distance de Manhattan)**.** *La distance de Manhattan (distance associée à la norme 1) entre deux observations* $x_i = (x_{i1}, \ldots, x_{ip})'$ *et* $x_{i'} = (x_{i'1}, \ldots, x_{i'p})'$ *de* $\mathbb{R}^p$ *est :*

$$d_{Man}(x_i, x_{i'}) = \sum_{j=1}^{p} |x_{ij} - x_{i'j}|.$$

6.4 Comment choisir la valeur K ?

Le choix de la valeur de K, c'est-à-dire le nombre de voisins à considérer pour prédire la sortie associée à une nouvelle observation, varie en fonction du jeu de données. En règle générale, moins on utilisera de voisins (un nombre K petit) plus on fera du sur-apprentissage (overfitting). Cela signifie que le modèle prédictif généré lors de la phase d'apprentissage est trop dépendant de l'ensemble d'apprentissage $\mathcal{D}$. Sa performance de prédiction sur ces données sera très bonne mais l'erreur (dite de généralisation) sur un nouvel ensemble de données risque d'être très mauvaise. Cela est dû au fait que le modèle prédictif "modélise" également des informations qui ne font pas partie de la véritable relation entrées-sortie, comme par exemple du bruit dû aux mesures.

Par ailleurs, plus on utilise de voisins (un nombre K grand) plus la prédiction sera stabilisée. Toutefois, si on utilise un nombre K trop grand (proche de n par exemple, n étant le nombre d'observations) on risque de faire face à un phénomène de sous-apprentissage (underfitting), autrement dit un modèle prédictif qui n'est pas assez souple pour rendre compte de la relation entrées-sortie.

Pour choisir K, on peut découper l'échantillon de données $\mathcal{D}$ en deux groupes. Le premier échantillon, appelé échantillon d'apprentissage, sert à construire la prédiction et avec le second échantillon, appelé échantillon de test, on estime les sorties et on évalue les performances. En faisant varier K, on regarde comment évoluent les performances d'estimation.

6.5 Exemple sur les données `iris`

Le jeu de données `iris` connu aussi sous le nom de "Iris de Fisher" est un jeu de données classiquement présenté en apprentissage. On dispose de 50 échantillons de chacune des trois espèces d'iris (Iris setosa, Iris virginica et Iris versicolor) pour lesquelles quatre caractéristiques ont été mesurées à partir de chaque échantillon : la longueur et la largeur des sépales et des pétales, en centimètres.

Comme pour les méthodes de classification non supervisée (K-means et CAH) du Chapitre 2, il est préférable de travailler sur des données centrées et réduites.

```
library(class)
data(iris)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
```

```
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

```
Npop<-nrow(iris)
taille.test<-Npop/3
K<-3

#création du découpage apprentissage-test
test.ind<-sample(1:Npop,taille.test)
appr.ind<-setdiff(1:Npop,test.ind)

#réalisation de la prédiction
knn.pred<-knn(iris[appr.ind,-5],iris[test.ind,-5],iris[appr.ind,5],k=K)
#matrice de confusion
table(iris[test.ind,5],knn.pred)
```

```
##             knn.pred
##              setosa versicolor virginica
##   setosa         12          0         0
##   versicolor      0         16         1
##   virginica       0          0        21
```

```
#taux d'erreur
mean(iris[test.ind,5]!=knn.pred)
```

```
## [1] 0.02
```

Que se passe-t-il si on choisit une autre valeur pour K ?

```
knn.bis<-knn(iris[appr.ind,-5],iris[test.ind,-5],iris[appr.ind,5],k=5)
#matrice de confusion
table(iris[test.ind,5],knn.bis)
```

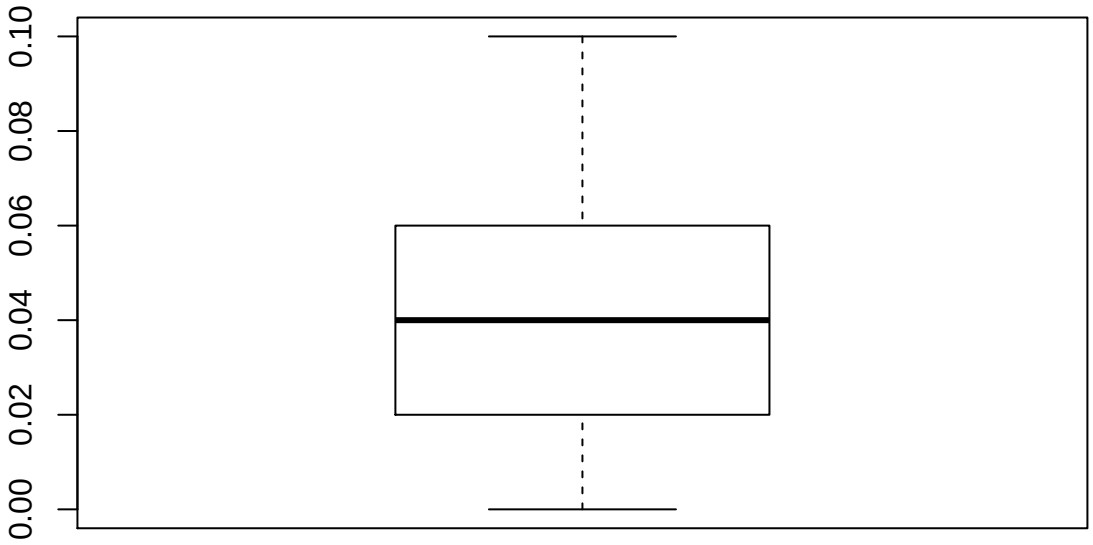
```
##             knn.bis
##              setosa versicolor virginica
##   setosa         12          0         0
##   versicolor      0         17         0
##   virginica       0          2        19
```

```r
#taux d'erreur
mean(iris[test.ind,5]!=knn.bis)
```

```
## [1] 0.04
```

On répète grâce à une boucle `for` le découpage des données et l'estimation de l'erreur de prédiction. Puis on représente les estimations obtenues avec une boîte à moustaches.

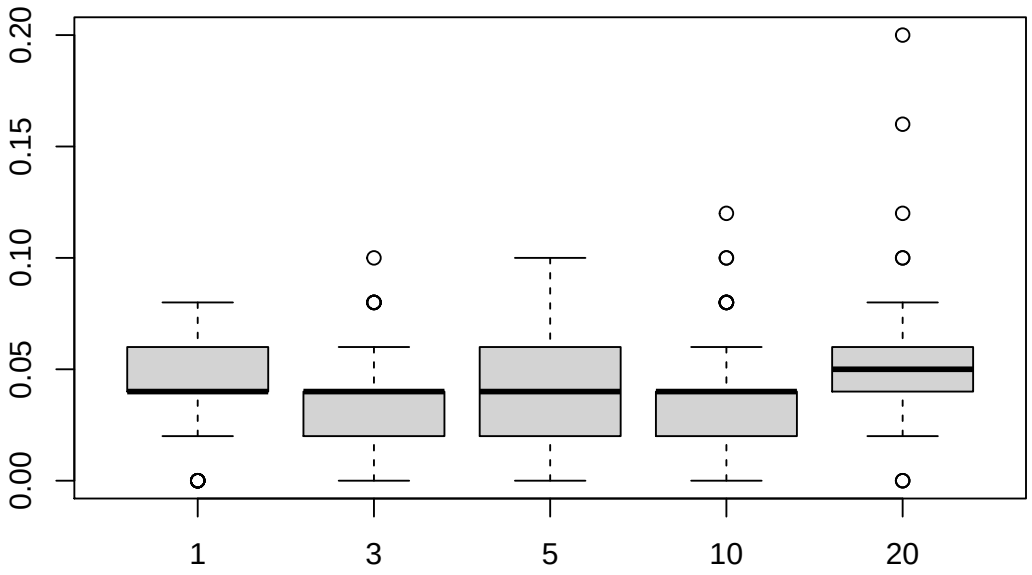
```r
err.at<-NULL
for(i in 1:100){
  test.ind<-sample(1:Npop,taille.test)
  appr.ind<-setdiff(1:Npop,test.ind)
  knn.pred<-knn(iris[appr.ind,-5],iris[test.ind,-5],iris[appr.ind,5],k=K)
  err.at[i]<-mean(iris[test.ind,5]!=knn.pred)}
boxplot(err.at)
```



Et on peut le faire pour plusieurs valeurs de K afin de choisir la valeur de K permettant de réaliser la meilleure prédiction :

```r
Kmin=1
Kmax=20
vK = c(seq(1,5,2),10,20)
imax=100
cpt = 1
err.at<-matrix(0,imax,length(vK))
for(K in vK){
  for(i in 1:imax){
    test.ind<-sample(1:Npop,taille.test)
    appr.ind<-setdiff(1:Npop,test.ind)
    knn.pred<-knn(iris[appr.ind,-5],iris[test.ind,-5],iris[appr.ind,5],k=K)
    err.at[i,cpt]<-mean(iris[test.ind,5]!=knn.pred)}
    cpt = cpt + 1
  }
```

```
boxplot(err.at, names = vK)
```



Cet exemple illustre le problème de la classification. La méthode est similaire pour le problème de la régression avec un calcul de moyenne (ou de médiane) à la place du vote majoritaire sur les K plus proches voisins. Il est à noter que la fonction `knn` de R ne fonctionne que pour le problème de classification. Pour réaliser une régression, on peut utiliser la fonction `knn.reg` du package `FNN`.