

# Régression en grande dimension

Janvier 2025

Amor KEZIOU  
Laboratoire de Mathématiques de Reims  
[amor.keziou@univ-reims.fr](mailto:amor.keziou@univ-reims.fr)

## Objectifs du module

- Présentation des différentes méthodes de validation croisée et de bootstrap ;
- Statistique en grande dimension ; maîtrise des méthodes de régularisation en régression (estimateurs Ridge, Lasso, ...) ;
- Savoir mettre en œuvre les différentes méthodes avec le logiciel R.

## Organisation du cours

- CM : 15h
- TD sur machine avec le logiciel R : 10h

## Évaluation

- Première session :
  - Une interrogation écrite (IE) d'une heure : 50% ;
  - Un projet (à réaliser par groupes de 3 ou 4 étudiants) : 50%.
- Deuxième session :
  - Un examen oral terminal (EOT) : 100%.

## Plan du cours

- Le problème de régression ;
- Erreur théorique (de prévision) d'un modèle de régression ;
- Méthodes de validation-croisée et bootstrap ;
- Sélection de modèles ;
- Méthodes de régularisation en régression (estimateurs ridge, lasso, ...) ;
- Méthodes d'estimation non paramétriques de la densité ;
- Références bibliographiques.

## Le problème de régression

- On cherche à expliquer/prédire une variable statistique  $Y$  à l'aide de  $p$  variables statistiques numériques  $X_1, \dots, X_p$  ;
- Notons  $\mathcal{X} \subset \mathbb{R}^p$  le domaine du vecteur  $\mathbf{X} := (X_1, \dots, X_p)$ , et  $\mathcal{Y}$  le domaine de  $Y$  ;
- Vocabulaire :
  - lorsque  $Y$  est quantitative “continue” ( $\mathcal{Y} \subset \mathbb{R}$ ), on parle de **modèle de régression** ;
  - lorsque  $Y$  est qualitative (ou quantitative) avec  $\text{Card}(\mathcal{Y})$  fini et “petit”, on parle d'**analyse discriminante**, ou **classification supervisée** ;
  - $Y$  est appelée : **la variable réponse**, la variable cible, la variable endogène, la variable à expliquer, la variable dépendante, ...
  - les variables  $X_1, \dots, X_p$  sont appelées : **les variables explicatives**, les variables exogènes, les variables “indépendantes”, les variables de contrôle, les régresseurs, les prédicteurs, ...

## Exemple 1

On cherche à expliquer/prédire la concentration en ozone ( $Y$ ), à l'aide de plusieurs variables explicatives :

- $X_1$  : température,
- $X_2$  : vitesse du vent,
- $X_3$  : jour de l'année,
- $X_4$  : humidité,
- $X_5$  : visibilité,
- $X_6, X_7, \dots$

## Questions

Comment expliquer (modéliser) la concentration en ozone ( $Y$ ) à l'aide de ces variables ? Quel est le modèle/le sous-ensemble de variables explicatives qui explique/prédit au "mieux" la concentration en ozone ? (sélection de modèles/sélection de variables).

## Le problème de régression :

- Il s'agit de trouver une fonction  $f : \mathbb{R}^p \mapsto \mathbb{R}$  telle que  $Y \approx f(X_1, \dots, X_p)$ ;
- Sauf dans des cas très particuliers, le lien n'est jamais parfait

$$Y = f(X_1, \dots, X_p) + \varepsilon.$$

## Les données :

- On dispose d'un  $n$ -échantillon i.i.d.  $(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)$  du vecteur aléatoire  $(\mathbf{X}, Y) \in \mathbb{R}^p \times \mathbb{R}$ . Notons les données  $D_n := \{(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)\}$ . L'objectif est alors de trouver la "**meilleure**" fonction  $\hat{f}_n(\cdot) : \mathbf{x} \in \mathbb{R}^p \mapsto \hat{f}_n(\mathbf{x}) := \hat{f}_n(\mathbf{x}; D_n)$ , fonction de  $\mathbf{x}$  et des données, telle que  $\hat{f}_n(\mathbf{X}_i) \approx Y_i, \forall i = 1, \dots, n, \quad \hat{f}_n(\mathbf{X}) \approx Y$ .

Cette fonction de prévision  $\hat{f}_n : \mathbf{x} \in \mathbb{R}^p \mapsto \mathbb{R}$  dépend des données  $D_n := \{(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)\} : \hat{f}_n(\mathbf{x}) := \hat{f}_n(\mathbf{x}; D_n)$ .

## Fonction de perte

- Nécessité de se donner un **critère** qui permet de mesurer la qualité des fonctions de prévision  $f$  ;
- Le plus souvent, on utilise une **fonction de perte**  $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$  telle que

$$\begin{cases} \ell(y, y') = 0 & \text{si } y = y' \\ \ell(y, y') > 0 & \text{si } y \neq y'. \end{cases}$$

## Risque théorique, ou erreur théorique (de test, de prévision)

- Étant donné une **fonction de perte**  $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ , la performance d'une fonction de prévision  $f : \mathcal{X} \rightarrow \mathcal{Y}$  est mesurée par

$$\mathcal{R}(f) := \mathbb{E} [\ell(Y, f(\mathbf{X}))],$$

où  $(\mathbf{X}, Y)$  est indépendant des  $(\mathbf{X}_i, Y_i)$ ,  $i = 1, \dots, n$ , et est de même loi  $P$ ;

- $\mathcal{R}(f)$  est appelé **risque théorique** ou **erreur théorique de prévision**, de  $f$ .

## Aspect théorique

- Pour une fonction de perte  $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ , le problème **théorique** consiste à trouver

$$f^* := \arg \min_f \mathcal{R}(f);$$

- Une telle fonction  $f^*$  (si elle existe) est appelée fonction **théorique** de prévision **optimale** pour la fonction de perte  $\ell$ ;
- La fonction de prévision optimale  $f^*$  dépend de la loi  $P$ , du vecteur aléatoire  $(\mathbf{X}, Y)$ , qui est en pratique **inconnue**;

- L'objectif est donc de trouver, à partir des données  $D_n$ , un **estimateur**  $\hat{f}_n : \mathbf{x} \in \mathcal{X} \mapsto \hat{f}_n(\mathbf{x}) := \hat{f}_n(\mathbf{x}; D_n) \in \mathcal{Y}$ , de la fonction de prévision  $f : \mathbf{x} \in \mathcal{X} \mapsto f(\mathbf{x}) \in \mathcal{Y}$ , de risque le plus faible possible :

$$\begin{aligned}\mathcal{R}(\hat{f}_n) &:= \mathbb{E} \left[ \ell \left( Y, \hat{f}_n(\mathbf{X}; D_n) \right) \right] \\ &\approx \mathbb{E} [\ell(Y, f^*(\mathbf{X}))] =: \mathcal{R}(f^*).\end{aligned}\tag{1}$$

## Définition : algorithme de prévision

- Un algorithme de prévision est représenté par une suite  $(\hat{f}_n)_n$  d'applications (mesurables) telles que pour tout  $n \geq 1$

$$\hat{f}_n : \mathcal{X} \times (\mathcal{X} \times \mathcal{Y})^n \rightarrow \mathcal{Y};$$

- On dit que la suite  $(\hat{f}_n)_n$  est **consistante** si

$$\lim_{n \rightarrow \infty} \mathcal{R}(\hat{f}_n) = \mathcal{R}(f^*).$$

## Choix de la fonction de perte

- Dans la définition précédente, la performance d'un estimateur de la fonction de prévision est définie vis-à-vis d'un **critère** (représenté par une fonction de perte  $\ell$ ) ;
- Un estimateur performant pour un critère ne sera pas forcément performant pour un autre ;
- **Conséquence** : avant de chercher à construire un estimateur, il est capital de savoir mesurer la performance (choisir la fonction de perte  $\ell$ ).

## Choix de la fonction de perte en régression

- En régression ( $\mathcal{Y} = \mathbb{R}$ ), la fonction de **perte quadratique** est la plus souvent utilisée. Elle est définie par

$$\begin{aligned}\ell : \mathbb{R} \times \mathbb{R} &\rightarrow \mathbb{R}_+ \\ (y, y') &\mapsto \ell(y, y') := (y - y')^2\end{aligned}$$

- Le **risque théorique** d'une fonction de prévision  $f : \mathcal{X} \rightarrow \mathbb{R}$  est donnée par (appelée **risque quadratique théorique**)

$$\mathcal{R}(f) := \mathbb{E} \left[ (Y - f(\mathbf{X}))^2 \right];$$

- La fonction de prévision optimale, dans ce cas, est la fonction **d'espérance conditionnelle** :

$$f^*(\cdot) : \mathbf{x} \in \mathbb{R}^p \mapsto f^*(\mathbf{x}) := \arg \min_f \mathcal{R}(f) = \mathbb{E}[Y | \mathbf{X} = \mathbf{x}].$$

- Cela signifie que, pour toute autre fonction de prévision  $f$ ,

$$\mathcal{R}(f^*) := \mathbb{E} \left[ (Y - f^*(\mathbf{X}))^2 \right] \leq \mathbb{E} \left[ (Y - f(\mathbf{X}))^2 \right] =: \mathcal{R}(f);$$

- **Problème** :  $f^*(\cdot)$  est inconnue en pratique. Il nous faut alors trouver, à partir des données  $D_n$ , un estimateur  $\hat{f}_n(\mathbf{x}) := \hat{f}_n(\mathbf{x}; D_n)$  tel que  $\hat{f}_n(\mathbf{x}) \approx f^*(\mathbf{x})$ .

## Proposition

- La suite d'estimateurs  $(\hat{f}_n(\mathbf{x}))_n$  est consistante si

$$\lim_{n \rightarrow +\infty} \mathbb{E} \left[ \int_{\mathcal{X}} \left( \hat{f}_n(\mathbf{x}; D_n) - f^*(\mathbf{x}) \right)^2 P_{\mathbf{X}}(d\mathbf{x}) \right] = 0.$$

## Autres choix de fonction de perte pour la régression

- La perte quadratique (ou  $L_2$ ) est un cas particulier des fonctions de pertes  $L_p$ ,  $p \geq 1$ ,

$$\begin{aligned}\ell_p &: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}_+ \\ (y, y') &\mapsto |y - y'|^p.\end{aligned}$$

- Pour le problème de régression, on privilégie souvent la perte quadratique, car moins difficile à minimiser et pour son interprétation en termes de **biais/variance**.
- Dans le cas de la perte  $L_1$ , on peut montrer que la fonction de prévision optimale est la fonction de la **médiane conditionnelle**

$$f^*(\cdot) : \mathbf{x} \in \mathbb{R}^p \mapsto f^*(\mathbf{x}) = \text{médiane}[Y \mid \mathbf{X} = \mathbf{x}].$$

On dispose de données  $D_n := \{(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)\}$  qu'on suppose i.i.d. de même loi que  $(\mathbf{X}, Y)$ , à valeur dans  $\mathcal{X} \times \mathcal{Y}$ .

## Objectif

- Étant donnée une fonction de perte  $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ , on cherche un estimateur  $\hat{f}_n(\cdot) := \hat{f}_n(\cdot; D_n)$ , de la fonction de prévision, qui soit le plus “proche” possible de la solution théorique optimale

$$f^* := \arg \min_f \mathcal{R}(f),$$

de sorte que  $\mathcal{R}(\hat{f}_n) := \mathbb{E} \left[ \ell \left( Y, \hat{f}_n(\mathbf{X}; D_n) \right) \right]$  soit minimal.

## Question :

- Étant donnée un estimateur  $\hat{f}_n(\cdot) := \hat{f}_n(\cdot; D_n)$  de la fonction de prévision, que vaut son risque  $\mathcal{R}(\hat{f}_n)$  ?
- La loi de  $(\mathbf{X}, Y)$  étant inconnue en pratique, il est impossible de calculer  $\mathcal{R}(\hat{f}_n) := \mathbb{E} [\ell(Y, \hat{f}_n(\mathbf{X}; D_n))]$ . Il faudrait donc l'estimer à partir des données. Comment ?
- Première approche :  $\mathcal{R}(\hat{f}_n)$  étant une espérance, on peut l'estimer par sa version empirique

$$\mathcal{R}_n(\hat{f}_n) := \frac{1}{n} \sum_{i=1}^n \ell(Y_i, \hat{f}_n(\mathbf{X}_i; D_n)).$$

Cette quantité s'appelle le risque empirique.

## Problème :

- L'échantillon  $D_n$  a été déjà utilisé pour construire l'estimateur  $\hat{f}_n(\cdot) := \hat{f}_n(\cdot; D_n)$ ; la loi des grands nombres ne peut donc s'appliquer !
- **Conséquence** : Le risque empirique  $\mathcal{R}_n(\hat{f}_n)$  conduit souvent à une **sous-estimation** du risque théorique de prévision  $\mathcal{R}(\hat{f}_n)$ .

## Solution :

Utiliser des méthodes de type **validation croisée** ou **bootstrap** (... à voir plus tard).

# Méthodes d'estimation de la fonction de prévision en régression

## Le cadre

- On dispose de données

$$D_n := \{(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)\}$$

qui sont supposées être i.i.d. de même loi que  $(\mathbf{X}, Y)$  à valeurs dans  $\mathcal{X} \times \mathcal{Y} = \mathbb{R}^p \times \mathbb{R}$ ;

- On considère la fonction de perte quadratique

$$\ell(\cdot, \cdot) : (y, y') \in \mathbb{R} \times \mathbb{R} \mapsto \ell(y, y') := (y - y')^2 \in \mathbb{R}_+.$$

## Objectif

- Construire, à partir des données  $D_n$ , un estimateur  $\hat{f}_n(\cdot) := \hat{f}_n(\cdot; D_n)$ , de la fonction de prévision, qui soit le plus proche possible de la fonction optimal

$$f^* := \arg \min_f \mathbb{E} \left[ (Y - f(\mathbf{X}))^2 \right],$$

de sorte que  $\mathcal{R}(\hat{f}_n)$  soit le plus petit possible.

## Modèles de régression

- Poser un modèle de régression revient à supposer que la fonction de prévision  $f(\cdot)$  appartient à un certain espace  $\mathcal{M}$ ;
- Le travail sera alors de trouver, à partir des données  $D_n$ , parmi les éléments  $f$  du modèle  $\mathcal{M}$ , la “meilleure” fonction  $\hat{f}_n(\cdot)$ .

## Modèles non paramétriques

- L'espace  $\mathcal{M}$  est de dimension infinie ;
- Exemple : On pose  $Y = f(X_1, \dots, X_p) + \varepsilon$ , où  $f$  appartient à l'espace des fonctions continues  $\mathbb{R}^p \rightarrow \mathbb{R}$ .

## Modèles paramétriques

- L'espace  $\mathcal{M}$  est de dimension finie ;
- Exemple : On suppose que la fonction de prévision optimale  $f^*(\mathbf{x}) := \mathbb{E}(Y | \mathbf{X} = \mathbf{x})$  est linéaire :

$$f^*(\mathbf{x}) = w_0 + w_1 x_1 + \cdots + w_p x_p;$$

- Cette modélisation est souvent réécrite sous la forme

$$Y = f(X_1, \dots, X_p; \mathbf{w}) + \varepsilon = w_0 + w_1 X_1 + \dots + w_p X_p + \varepsilon, \quad (2)$$

avec  $\mathbb{E} [\varepsilon | \mathbf{X} = \mathbf{x}] = 0$  et  $\text{Var} [\varepsilon | \mathbf{X} = \mathbf{x}] = \sigma^2$ , ne dépendant pas de  $\mathbf{x}$  (hypothèse d'**homoscédasticité**).

- Le problème est alors d'estimer le paramètre  $\mathbf{w} := (w_0, w_1, \dots, w_p)^\top \in \mathbb{R}^{1+p}$  à l'aide des données  $(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)$ ;
- Le modèle ci-dessus s'appelle **le modèle de régression linéaire**.

## Remarque

- Intuitivement, une modélisation **non-paramétrique** a l'air plus attractive (car plus **flexible**) ;
- Il y a cependant un prix à payer ... Il est en général plus difficile de "bien estimer" dans un modèle non-paramétrique, en particulier lorsque le nombre de variables explicatives est grand.

On considère d'abord le cas plus simple d'un modèle de régression linéaire simple (RLS), i.e.,

$$Y = w_0 + w_1 X + \varepsilon,$$

où  $X$  est réelle,  $\mathbb{E}[\varepsilon | X = x] = 0$ , et  $\text{Var}[\varepsilon | X = x] = \sigma^2$ , ne dépendant pas de  $x$  (hypothèse d'**homoscédasticité**).

## Estimation par moindres carrés

- Elle consiste à minimiser en  $(w_0, w_1) \in \mathbb{R}^2$  la moyenne des carrés des erreurs

$$\frac{1}{n} \sum_{i=1}^n \varepsilon_i^2 := \frac{1}{n} \sum_{i=1}^n (Y_i - w_0 - w_1 X_i)^2,$$

qui fournit les estimateurs

$$\widehat{w}_0 = \overline{Y} - \widehat{w}_1 \overline{X}; \quad \widehat{w}_1 = \frac{\sum_{i=1}^n (Y_i - \overline{Y})(X_i - \overline{X})}{\sum_{i=1}^n (X_i - \overline{X})^2}.$$

## Propriétés des estimateurs MC

### Sous les conditions

- Les erreurs  $\varepsilon_i, i = 1, \dots, n$ , sont non corrélées,
- $\mathbb{E}(\varepsilon_i | X = x) = 0$ ,  $\text{Var}(\varepsilon_i | X = x) = \sigma^2, i = 1, \dots, n$ , (ne dépendant pas de  $x$  : hypothèse d'homoscédasticité),

on a les propriétés suivantes

- $\hat{w}_0$  et  $\hat{w}_1$  sont des estimateurs sans biais de  $w_0$  et  $w_1$  ;
- Les variances (conditionnellement aux  $X_1, \dots, X_n$ ) sont données par

$$\text{Var}(\hat{w}_0 | X_1, \dots, X_n) = \sigma^2 \left( \frac{1}{n} + \frac{\bar{X}^2}{\sum_{i=1}^n (X_i - \bar{X})^2} \right),$$

$$\text{Var}(\hat{w}_1 | X_1, \dots, X_n) = \frac{\sigma^2}{\sum_{i=1}^n (X_i - \bar{X})^2}.$$

## Vocabulaire

- $\hat{w}_0 + \hat{w}_1 X_i =: \hat{Y}_i$  s'appelle la **valeur ajustée** de l'observation  $Y_i$  ;
- $\hat{\varepsilon}_i := Y_i - \hat{Y}_i$  s'appelle **résidu** ;
- Étant donné  $X_{n+1}$  une nouvelle valeur de la variable  $X$  ; on cherche à estimer la valeur  $Y_{n+1} := w_0 + w_1 X_{n+1}$  ; un estimateur naturel est la **prévision** associée

$$\hat{Y}_{n+1} := \hat{w}_0 + \hat{w}_1 X_{n+1}.$$

- Cette valeur s'appelle la **valeur prédictive** de  $Y$  (donnée par le modèle) quand  $X = X_{n+1}$ .

## Propriété

On a

$$\text{Var}(\hat{Y}_{n+1} | X_1, \dots, X_n, X_{n+1}) = \sigma^2 \left( \frac{1}{n} + \frac{(X_{n+1} - \bar{X})^2}{\sum_{i=1}^n (X_i - \bar{X})^2} \right).$$

Cette variance de la prévision est d'autant plus faible que

- $\sigma^2$  est petit ;
- $X_{n+1}$  est proche du centre de gravité  $\bar{X}$  des  $X_i$ . (C'est plus difficile de bien prédire vers les extrêmes).

## Questions

- Comment mesurer la qualité du modèle ?
- Comment tester la valeur des coefficients du modèle ?
- Peut-on obtenir des intervalles de confiance pour les paramètres  $w_j$  ou pour la prévision  $\hat{Y}_{n+1}$  ?

## Condition

On suppose que les erreurs  $\varepsilon_i, i = 1, \dots, n$ , sont i.i.d. (et indépendantes de  $X_1, \dots, X_n$ ) de même loi normale  $\mathcal{N}(0, \sigma^2)$ , de variance  $\sigma^2$  (ne dépendant pas de  $x$ ).

Si cette condition est vérifiée, alors on a

### Cas 1 : si $\sigma^2$ connue

- $\mathcal{L}(\hat{w}_0 | X_1, \dots, X_n) = \mathcal{N}(w_0, \text{Var}(\hat{w}_0 | X_1, \dots, X_n))$  et  $\mathcal{L}(\hat{w}_1 | X_1, \dots, X_n) = \mathcal{N}(w_1, \text{Var}(\hat{w}_1 | X_1, \dots, X_n))$ .

### Cas 2 : si $\sigma^2$ inconnue

On estime  $\sigma^2$  par  $\hat{\sigma}^2 := \frac{1}{n-2} \sum_{i=1}^n \hat{\varepsilon}_i^2$ , et on a

- $(n-2) \frac{\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{(n-2)}$ , (une loi du  $\chi^2$  à  $n-2$  ddl) ;
- $(\hat{w}_0, \hat{w}_1)$  et  $\hat{\sigma}^2$  sont indépendants ;
- $\frac{\hat{w}_0 - w_0}{\hat{\sigma}_{\hat{w}_0}} \sim \mathcal{T}_{(n-2)}$ , et  $\frac{\hat{w}_1 - w_1}{\hat{\sigma}_{\hat{w}_1}} \sim \mathcal{T}_{(n-2)}$ ,  $\mathcal{T}_{(n-2)}$  étant la loi de Student à  $(n-2)$  ddl,

$$\widehat{\sigma}_{\widehat{w}_0} := \sqrt{\text{Var}(\widehat{w}_0 | X_1, \dots, X_n)} = \sqrt{\widehat{\sigma}^2 \left( \frac{1}{n} + \frac{\overline{X}^2}{\sum_{i=1}^n (X_i - \overline{X})^2} \right)},$$

et

$$\widehat{\sigma}_{\widehat{w}_1} := \sqrt{\text{Var}(\widehat{w}_1 | X_1, \dots, X_n)} = \sqrt{\frac{\widehat{\sigma}^2}{\sum_{i=1}^n (X_i - \overline{X})^2}}.$$

Au vu des résultats précédents, on obtient les intervalles de confiances suivants, au niveau  $(1 - \alpha)$ , des paramètres  $w_0$  et  $w_1$  :

### Intervalles de confiance pour les paramètres $w_0$ et $w_1$

- Pour  $w_0$  :

$$[\hat{w}_0 - t_{(n-2)}(1 - \alpha/2) \hat{\sigma}_{\hat{w}_0}, \hat{w}_0 + t_{(n-2)}(1 - \alpha/2) \hat{\sigma}_{\hat{w}_0}] ,$$

où  $t_{(n-2)}(1 - \alpha/2)$  est le quantile d'ordre  $(1 - \alpha/2)$  de la loi de Student à  $(n - 2)$  ddl ;

- Pour  $w_1$  :

$$[\hat{w}_1 - t_{(n-2)}(1 - \alpha/2) \hat{\sigma}_{\hat{w}_1}, \hat{w}_1 + t_{(n-2)}(1 - \alpha/2) \hat{\sigma}_{\hat{w}_1}] .$$

## Intervalles d'estimation et de prévision d'une valeur prédictive

- Intervalle de confiance d'**estimation** pour  $Y_{n+1}$

$$\left[ \hat{Y}_{n+1} \pm t_{(n-2)}(1 - \alpha/2) \hat{\sigma} \sqrt{\frac{1}{n} + \frac{(X_{n+1} - \bar{X})^2}{\sum_{i=1}^n (X_i - \bar{X})^2}} \right];$$

- Intervalle de confiance de **prévision** pour  $Y_{n+1}$

$$\left[ \hat{Y}_{n+1} \pm t_{(n-2)}(1 - \alpha/2) \hat{\sigma} \sqrt{1 + \frac{1}{n} + \frac{(X_{n+1} - \bar{X})^2}{\sum_{i=1}^n (X_i - \bar{X})^2}} \right].$$

- Pour le dernier intervalle, on tient compte également de la variabilité des résidus.

On souhaite tester l'hypothèse nulle

$$\mathcal{H}_0 : w_1 = 0 \quad \text{contre l'alternative} \quad \mathcal{H}_1 : w_1 \neq 0.$$

On utilise la statistique (de Student)  $t := \frac{\hat{w}_1}{\hat{\sigma}_{\hat{w}_1}}$  qui suit une loi de Student à  $(n - 2)$  ddl si  $\mathcal{H}_0$  est vraie.

La  $P$ -value de ce test est alors définie par

$$P\text{-value} := \mathbb{P}(|T| > |t_{obs}|),$$

où  $T$  est une v.a. suivant la loi de Student à  $(n - 2)$  ddl. On utilise alors la règle de décision suivante

- on rejette  $\mathcal{H}_0$  si la  $P$ -value  $< \alpha$ ;
- on accepte  $\mathcal{H}_0$  si la  $P$ -value  $\geq \alpha$ ,

$\alpha$  étant le risque de première espèce, i.e., la probabilité de rejeter  $\mathcal{H}_0$  alors qu'elle est vraie (cette valeur est prise souvent dans la pratique égale à 0.01 ou 0.05).

## Retour à l'exemple 1

- On cherche à expliquer/prédire la concentration en ozone ( $Y$ ) à l'aide de 9 variables explicatives possibles :  $X_1$  : température ;  $X_2$  : vitesse du vent ;  $X_3$  : ... ; ...
- On dispose de  $n = 330$  mesures (observations) du vecteur  $(\mathbf{X}, Y) := (X_1, \dots, X_9, Y)$  ; c.f. la base de données dénommée "LAozoneData" disponible en ligne sur Moodle en format ".xlsx" ; voici les 6 premières lignes de cette base

```
library(xlsx)
LAozoneData <- read.xlsx("LAozoneData.xlsx", sheetIndex = 1, header = T)
head(LAozoneData, 6)

##   ozone    vh wind humidity temp ibh dpg ibt vis doy
## 1     3 5710     4      28    40 2693 -25   87 250   3
## 2     5 5700     3      37    45 590 -24 128 100   4
## 3     5 5760     3      51    54 1450  25 139 60   5
## 4     6 5720     4      69    35 1568  15 121 60   6
## 5     4 5790     6      19    45 2631 -33 123 100   7
## 6     4 5790     3      25    55 554 -28 182 250   8
```

Figure – La base de données “LAozoneData”

Nous allons, dans un premier temps, utiliser seulement la variable explicative “X : temperature” pour expliquer/prédire la variable “Y : concentration en ozone” à l'aide d'un modèle de régression linéaire (simple). En R, on utilise la fonction “`lm()`” ou “`glm()`”.

Représentons d'abord les  $n = 330$  observations dans le plan à l'aide des deux variables  $(X, Y) \in \mathbb{R}^2$  :

## Exemple de la concentration en ozone

○○●○○○

```
library(ggplot2)
ggplot(data = LAozoneData, mapping = aes(x = temp, y = ozone)) + geom_point()
```

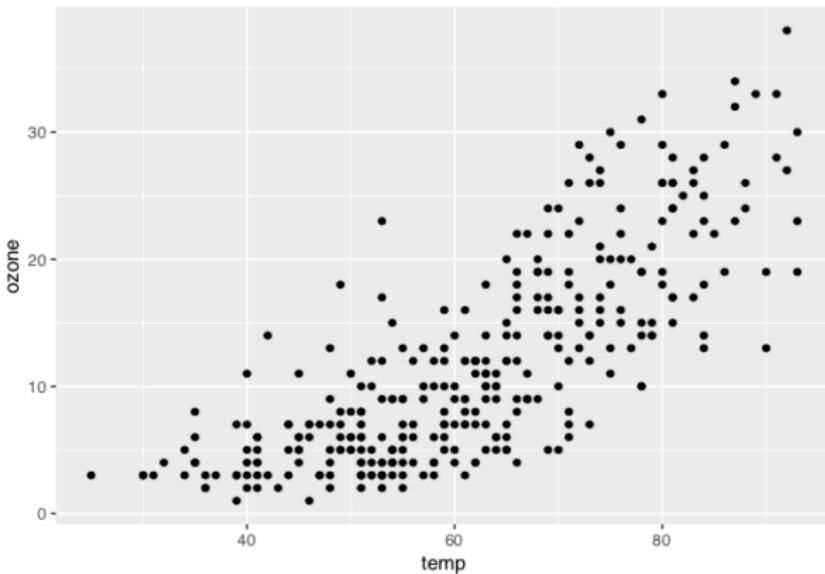


Figure – Nuage des points : ozone versus température

```
ggplot(data = LAozoneData, mapping = aes(x = temp, y = ozone)) + geom_point() +  
  geom_smooth(method = lm, se = FALSE)
```

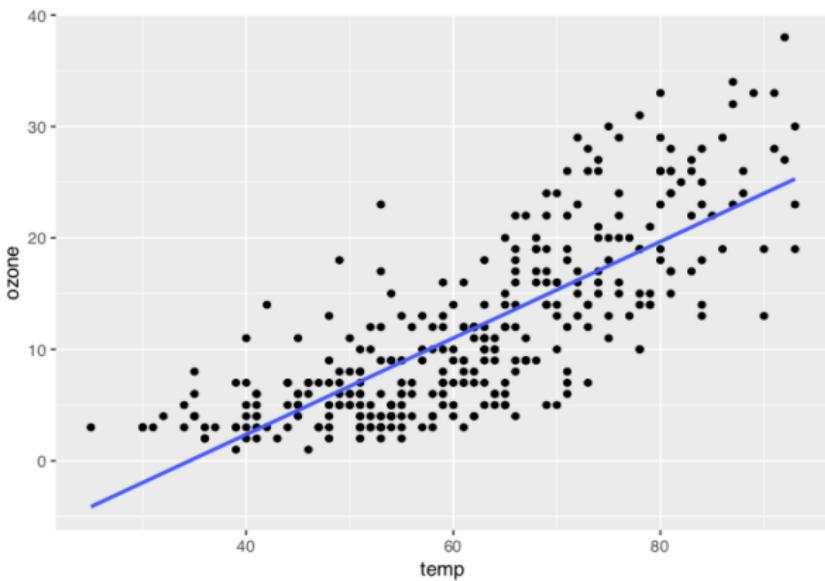


Figure – Ajustement par une droite

On construit maintenant le modèle de régression linéaire simple de la variable 'ozone' en fonction de la variable 'temp' :

```
modele.reg.simple <- lm(formula = ozone ~ temp, data = LAozoneData)
summary(modele.reg.simple)

##
## Call:
## lm(formula = ozone ~ temp, data = LAozoneData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -10.9939 -3.8202 -0.1796  3.1951 15.0112 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -14.93745   1.21247 -12.32   <2e-16 ***
## temp         0.43257   0.01912   22.63   <2e-16 ***
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 5.014 on 328 degrees of freedom
## Multiple R-squared:  0.6095, Adjusted R-squared:  0.6083 
## F-statistic: 511.9 on 1 and 328 DF,  p-value: < 2.2e-16
```

Figure – Résultats du modèle de régression linéaire simple

On trace ici la bande de confiance d'estimation

```
intervalles.estimation <- predict(modele.reg.simple, interv = "confidence",
                                    newdata = data.frame(temp = sort(LA ozoneData$temp)))
plot(LA ozoneData$ozone ~ LA ozoneData$temp, xlab = "temp", ylab = "ozone")
matplot(sort(LA ozoneData$temp), intervalles.estimation, type = "l",
        lty = c(1,2,2), add = T, col = c("black", "red", "red"), lwd = c(3,3,3))
```

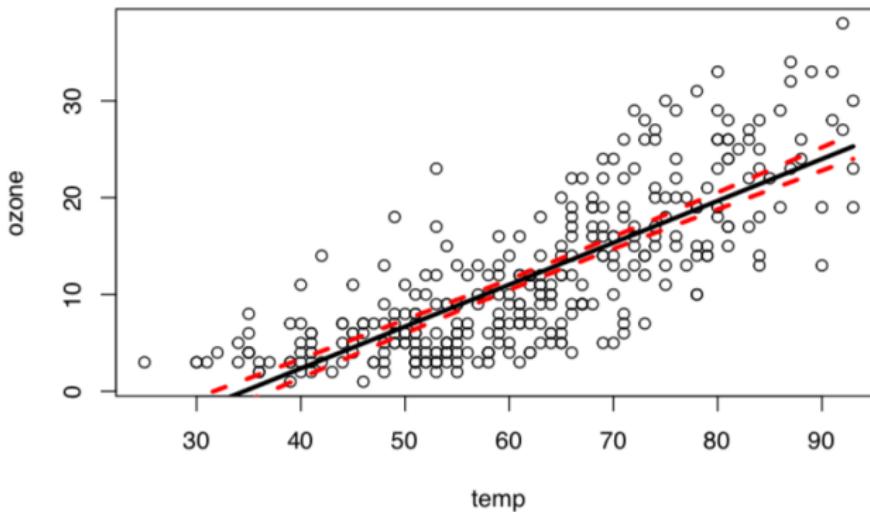


Figure – Intervalles de confiance d'estimation des valeurs ajustées

Ici on trace la bande de confiance de prévision

```
intervalles.prevision <- predict(modele.reg.simple, interv = "prediction",
                                 newdata = data.frame(temp = sort(LAozoneData$temp)))
plot(LAozoneData$ozone ~ LAozoneData$temp, xlab = "temp", ylab = "ozone")
matplot(sort(LAozoneData$temp), intervalles.prevision, type = "l",
        lty = c(1,2,2), add = T, col = c("black", "red", "red"), lwd = c(3,3,3))
```

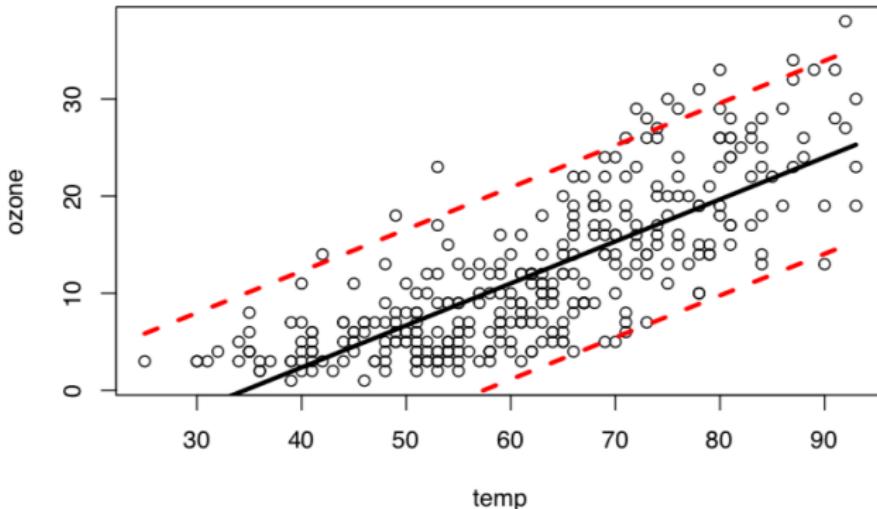


Figure – Intervalles de confiance de prévision des valeurs ajustées

# La régression linéaire multiple (RLM)

## Retour à l'exemple 1

- La température n'est pas la seule variable permettant d'expliquer ou de prédire la concentration en ozone ;
- D'autres variables peuvent être prise en compte (humidité, visibilité, jour de l'année, ...);
- D'où la nécessité d'étendre le modèle linéaire simple à plus d'une seule variable explicative.

## Notations

- $Y$  : variable statistique réelle (à expliquer/prédire) ;
- $\mathbf{X} := (X_1, \dots, X_p) \in \mathbb{R}^P$  : le vecteur, des variables explicatives, à valeurs dans  $\mathbb{R}^P$  ;
- On dispose de  $n$  observations  $(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)$  i.i.d. de même loi que  $(\mathbf{X}, Y) \in \mathbb{R}^P \times \mathbb{R}$ .

## Le modèle de régression linéaire multiple

Il s'écrit

$$Y = w_0 + w_1 X_1 + \dots + w_p X_p + \varepsilon,$$

où  $\mathbb{E}(\varepsilon | \mathbf{X} = \mathbf{x}) = 0$ , et  $\text{Var}(\varepsilon | \mathbf{X} = \mathbf{x}) = \sigma^2$ , ne dépendant pas de  $\mathbf{x}$  (hypothèse d'**homoscédasticité**).

## Notation matricielle

- On note

$$\mathbf{Y} := \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} \in \mathbb{R}^n, \quad \mathbb{X} := \begin{bmatrix} 1 & X_{1,1} & \cdots & X_{1,p} \\ \vdots & \vdots & & \vdots \\ 1 & X_{n,1} & \cdots & X_{n,p} \end{bmatrix} \in \mathcal{M}_{n \times (1+p)}(\mathbb{R}),$$

$$\mathbf{w} := \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix} \in \mathbb{R}^{1+p}, \quad \boldsymbol{\varepsilon} := \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix} \in \mathbb{R}^n,$$

qui représentent, respectivement, le vecteur des valeurs observées de la variable réponse  $Y$ , la **matrice de design**, le vecteur des paramètres du modèle, et enfin le vecteur des termes d'erreur du modèle,  $\varepsilon_1, \dots, \varepsilon_n$ , **non observables**.

Le modèle de RLM, associé aux données, s'écrit donc sous la forme matricielle suivante

$$\mathbf{Y} = \mathbb{X} \mathbf{w} + \boldsymbol{\varepsilon}.$$

## Définition

On appelle **estimateur des moindres carrés**  $\hat{\mathbf{w}}$ , de  $\mathbf{w} := (w_0, w_1, \dots, w_p)^\top$ , la statistique suivante

$$\begin{aligned}\hat{\mathbf{w}} &:= \arg \min_{\mathbf{w} \in \mathbb{R}^{1+p}} \frac{1}{n} \sum_{i=1}^n (Y_i - w_0 - w_1 X_{i,1} - \cdots - w_p X_{i,p})^2 \\ &= \arg \min_{\mathbf{w} \in \mathbb{R}^{1+p}} \frac{1}{n} \|\mathbf{Y} - \mathbb{X} \mathbf{w}\|^2.\end{aligned}$$

## Proposition

Si la matrice de design  $\mathbb{X}$  est de rang  $1 + p$  (ce qui nécessite  $n > p$ ), alors l'estimateur des MC, de  $\mathbf{w}$ , existe et est unique et est donné par

$$\hat{\mathbf{w}} := (\mathbb{X}^\top \mathbb{X})^{-1} \mathbb{X}^\top \mathbf{Y}.$$

## Estimation des paramètres du modèle RLM, par maximum de vraisemblance (MV)

On suppose que les termes d'erreur,

$$\varepsilon_1, \dots, \varepsilon_n,$$

sont i.i.d. de même loi normale  $\mathcal{N}(0, \sigma^2)$ ,  $\mathbb{E}(\varepsilon | \mathbf{X} = \mathbf{x}) = 0$ , et  $\text{Var}(\varepsilon | \mathbf{X} = \mathbf{x}) = \sigma^2$  ne dépendant pas de  $\mathbf{x}$  (**hypothèse d'homoscédasticité**).

Par conséquent, la loi conditionnelle de  $Y$ , sachant  $\mathbf{X} = \mathbf{x}$ , est une loi normale  $\mathcal{N}(w_0 + w_1x_1 + \dots + w_px_p, \sigma^2)$ , de variance  $\sigma^2$  ne dépendant pas de  $\mathbf{x}$ . La log-vraisemblance (conditionnelle) s'écrit alors

$$\mathcal{L}_n(\mathbf{w}, \sigma^2) = -\frac{n}{2} \log \sigma^2 - \frac{n}{2} \log(2\pi) - \frac{1}{2\sigma^2} \|\mathbf{Y} - \mathbb{X}\mathbf{w}\|^2.$$

On peut voir facilement que l'estimateur du MV, de  $\mathbf{w}$ , coïncide avec l'estimateur des MC.

En effet, notons  $(\tilde{\mathbf{w}}, \tilde{\sigma}^2)$  l'EMV de  $(\mathbf{w}, \sigma^2)$ , i.e.,

$$(\tilde{\mathbf{w}}, \tilde{\sigma}^2) := \arg \max_{\mathbf{w}, \sigma^2} \mathcal{L}_n(\mathbf{w}, \sigma^2).$$

Il est clair que l'EMV

$$\tilde{\mathbf{w}} = \arg \min_{\mathbf{w}} \|\mathbf{Y} - \mathbb{X}\mathbf{w}\|^2 =: \hat{\mathbf{w}} =: EMC.$$

La log-vraisemblance du modèle (évaluée à l'EMV) vaut

$$\mathcal{L}_n(\tilde{\mathbf{w}}, \tilde{\sigma}^2) = -\frac{n}{2} \log \frac{\|\hat{\boldsymbol{\varepsilon}}\|^2}{n} - \frac{n}{2}(1 + \log(2\pi)). \quad (3)$$

## Propriétés des estimateurs MC

Sous les conditions

- Les erreurs  $\varepsilon_i, i = 1, \dots, n$ , sont non corrélées,
- $\mathbb{E}(\varepsilon_i | \mathbb{X}) = 0$ ,  $\text{Var}(\varepsilon_i | \mathbb{X}) = \sigma^2, i = 1, \dots, n$ ,

on a les propriétés suivantes

- $\hat{\mathbf{w}}$  est un estimateur sans biais de  $\mathbf{w}$  ;
- La matrice de variance-covariance de  $\hat{\mathbf{w}}$ , conditionnellement à  $\mathbb{X}$ , est donnée par

$$\text{Var}(\hat{\mathbf{w}} | \mathbb{X}) = \sigma^2 (\mathbb{X}^\top \mathbb{X})^{-1}.$$

## Lois des estimateurs MC

On suppose que le vecteur des erreurs  $\varepsilon$  est indépendant de  $\mathbb{X}$ , et que  $\varepsilon \sim \mathcal{N}_n(\mathbf{0}, \sigma^2 \mathbb{I}_n)$ , une loi gaussienne sur  $\mathbb{R}^n$  centrée de matrice de variance-covariance  $\sigma^2 \mathbb{I}_n$ .

- Soit  $\hat{\varepsilon} := \mathbf{Y} - \mathbb{X} \hat{\mathbf{w}} =: \mathbf{Y} - \hat{\mathbf{Y}}$  le vecteur des résidus, et  $\hat{\sigma}^2$  l'estimateur, de  $\sigma^2$ , définie par

$$\hat{\sigma}^2 := \frac{1}{n - (1 + p)} \|\hat{\varepsilon}\|^2 = \frac{1}{n - (1 + p)} \sum_{i=1}^n \hat{\varepsilon}_i^2.$$

## Proposition

- $\hat{\mathbf{w}}$  suit, conditionnellement à  $\mathbb{X}$ , une loi gaussienne d'espérance  $\mathbf{w}$  et de matrice de variance-covariance  $\sigma^2 (\mathbb{X}^\top \mathbb{X})^{-1}$ ;
- $(n - p - 1) \frac{\hat{\sigma}^2}{\sigma^2} \sim \chi_{(n-p-1)}^2$ ;
- $\hat{\mathbf{w}}$  et  $\hat{\sigma}^2$  sont indépendants.

## Intervalles de confiance et tests

On note  $\hat{\sigma}_j^2 := \hat{\sigma}^2 \left[ (\mathbb{X}^\top \mathbb{X})^{-1} \right]_{j,j}$ , pour  $j = 0, 1, \dots, p$ . On a

$$\forall j = 0, \dots, p, \quad \frac{\hat{w}_j - w_j}{\hat{\sigma}_j} \sim \mathcal{T}_{(n-p-1)},$$

ce qui permet de construire des intervalles de confiance, au niveau  $1 - \alpha$ , pour les paramètres  $w_j$ , et de réaliser des tests d'hypothèses du type  $\mathcal{H}_0 : w_j = 0$  contre  $\mathcal{H}_1 : w_j \neq 0$ .

Intervalle de confiance, au niveau  $1 - \alpha$ , pour  $w_j$

$$[\hat{w}_j - t_{(n-p-1)}(1 - \alpha/2) \hat{\sigma}_j, \hat{w}_j + t_{(n-p-1)}(1 - \alpha/2) \hat{\sigma}_j].$$

### Test de Student

Considérons le problème de test de l'hypothèse

$$\mathcal{H}_0 : w_j = 0 \text{ contre l'alternative } \mathcal{H}_1 : w_j \neq 0.$$

Soit  $t := \frac{\hat{w}_j}{\hat{\sigma}_j}$  (la statistique de Student). La  $P$ -value du test (de Student) est donnée par

$$P\text{-value} = \mathbb{P}(|T| > |t_{obs}|),$$

où  $T$  est une variable aléatoire suivant la loi  $\mathcal{T}_{(n-p-1)}$ .

## Prévision et intervalles de confiance

- On dispose d'une nouvelle observation  $\mathbb{X}_{n+1} := (1, X_{n+1,1}, \dots, X_{n+1,p})$ . On prédit la valeur  $Y_{n+1}$  correspondante par
$$\hat{Y}_{n+1} := \mathbb{X}_{n+1} \hat{\mathbf{w}}.$$
- Intervalle d'**estimation**, au niveau  $1 - \alpha$ , pour  $Y_{n+1}$ 
$$\left[ \hat{Y}_{n+1} \begin{array}{c} + \\ - \end{array} t_{(n-p-1)}(1 - \alpha/2) \hat{\sigma} \sqrt{\mathbb{X}_{n+1} (\mathbb{X}^\top \mathbb{X})^{-1} \mathbb{X}_{n+1}^\top} \right];$$
- Intervalle de **prévision**, au niveau  $1 - \alpha$ , pour  $Y_{n+1}$ 
$$\left[ \hat{Y}_{n+1} \begin{array}{c} + \\ - \end{array} t_{(n-p-1)}(1 - \alpha/2) \hat{\sigma} \sqrt{1 + \mathbb{X}_{n+1} (\mathbb{X}^\top \mathbb{X})^{-1} \mathbb{X}_{n+1}^\top} \right].$$

## Retour à l'exemple 1

- On cherche à expliquer/prédire la concentration en ozone  $Y$  à l'aide de 9 variables explicatives possibles :  $X_1$  : température ;  $X_2$  : vitesse du vent ;  $X_3$  : ... ; ...
- On dispose de  $n = 330$  mesures (observations) du vecteur  $(X_1, \dots, X_9, Y)$ .

Nous allons utiliser toutes les variables explicatives  $(X_1, \dots, X_9) =: \mathbf{X}$  pour expliquer/prédire la variable réponse “ $Y$  : concentration en ozone”, à l'aide d'un modèle de régression linéaire multiple. La base de données “LAozoneData” est constituée de  $n = 330$  observations du vecteur aléatoire  $(\mathbf{X}, Y)$  :

Voici la structure de la base de données “LAozoneData”

```
str(LAozoneData)

## 'data.frame':    330 obs. of  10 variables:
## $ ozone     : num  3 5 5 6 4 4 6 7 4 6 ...
## $ vh        : num  5710 5700 5760 5720 5790 5790 5700 5700 5770 5720 ...
## $ wind       : num  4 3 3 4 6 3 3 3 8 3 ...
## $ humidity   : num  28 37 51 69 19 25 73 59 27 44 ...
## $ temp       : num  40 45 54 35 45 55 41 44 54 51 ...
## $ ibh        : num  2693 590 1450 1568 2631 ...
## $ dpg        : num  -25 -24 25 15 -33 -28 23 -2 -19 9 ...
## $ ibt        : num  87 128 139 121 123 182 114 91 92 173 ...
## $ vis         : num  250 100 60 60 100 250 120 120 120 150 ...
## $ doy        : num  3 4 5 6 7 8 9 10 11 12 ...
```

Figure – Structure de la base de données “LAozoneData”

## On construit ici le modèle de RLM

```

modele.reg.multiple <- lm(formula = ozone ~ ., data = LAozoneData)
summary(modele.reg.multiple)

##
## Call:
## lm(formula = ozone ~ ., data = LAozoneData)
##
## Residuals:
##      Min    1Q   Median    3Q   Max 
## -12.2407 -2.8832 -0.3353  2.7409 13.3523
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 18.3792938 29.5045242  0.623  0.53377  
## vh          -0.0051340  0.0053950 -0.952  0.34200  
## wind        -0.0198304  0.1238829 -0.160  0.87292  
## humidity     0.0804923  0.0188345  4.274 2.54e-05 ***
## temp         0.2743349  0.0497361  5.516 7.17e-08 ***
## ibh         -0.0002497  0.0002950 -0.846  0.39798  
## dpg         -0.0036968  0.0112925 -0.327  0.74360  
## ibt          0.0292640  0.0136115  2.150  0.03231 *  
## vis          -0.0080742  0.0037565 -2.149  0.03235 *  
## doy          -0.0088490  0.0027199 -3.253  0.00126 ** 
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.441 on 320 degrees of freedom
## Multiple R-squared:  0.7011, Adjusted R-squared:  0.6927 
## F-statistic: 83.4 on 9 and 320 DF,  p-value: < 2.2e-16

```

Figure – Résultats du modèle de RLM

## Questions

- Le modèle linéaire repose sur certaines hypothèses (non corrélation des termes d'erreur, relation linéaire entre la variable réponse et les variables explicatives, homoscédasticité, indépendance des erreurs, normalité des erreurs, ... ).  
Comment les vérifier ?
- Analyse des résidus ; détection des **outliers** (valeurs aberrantes de la variable réponse), détection des **points leviers extrêmes** (valeurs extrêmes des variables explicatives), détection des **points trop influents**, ...
- À partir de  $p$  variables explicatives, il est possible de construire  $2^p - 1$  modèles de régression linéaires possibles ; comment choisir le “meilleur” sous-ensemble de variables explicatives à inclure dans le modèle ? (sélection de modèles, de variables, ...), détection des points **trop influents** (qui peuvent être des outliers ou points levier extrêmes),

## Analyse des résidus

Test de non corrélation des erreurs en régression : **test de Durbin-Watson**

On cherche à tester l'hypothèse nulle

$\mathcal{H}_0$  : les erreurs  $\varepsilon_1, \dots, \varepsilon_n$  sont non corrélées

contre l'hypothèse alternative

$\mathcal{H}_1$  : les erreurs  $\varepsilon_1, \dots, \varepsilon_n$  sont corrélées.

- Les erreurs  $(\varepsilon_1, \dots, \varepsilon_n)$  ne sont pas observables ; elles sont “estimées” par le vecteur des résidus

$$\hat{\boldsymbol{\varepsilon}} := (\hat{\varepsilon}_1, \dots, \hat{\varepsilon}_n)^\top := \mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{Y} - \mathbb{X} \hat{\mathbf{w}}.$$

## Les résidus vérifient les propriétés suivantes

- $\sum_{i=1}^n \hat{\varepsilon}_i = 0, \quad \hat{\varepsilon}^\top \mathbb{X} := (\mathbf{Y} - \mathbb{X} \hat{\mathbf{w}})^\top \mathbb{X} = \mathbf{0};$
- $\forall i = 1, \dots, n, \quad \mathbb{E}(\hat{\varepsilon}_i) = 0;$
- Sous les hypothèses d'homoscédasticité, et de non corrélation des erreurs  $\varepsilon_i$ , on a  $\text{Var}(\hat{\varepsilon} | \mathbb{X}) = \sigma^2 (\mathbb{I}_n - \mathbb{X} (\mathbb{X}^\top \mathbb{X})^{-1} \mathbb{X}^\top).$

Le test de Durbin-Watson suppose le lien (modèle auto-régressif d'ordre 1) suivant entre les erreurs

$$\varepsilon_i = \rho \varepsilon_{i-1} + u_i, \quad i = 2, \dots, n,$$

où  $u_2, \dots, u_n$  est un **bruit blanc**, i.e., une suite de v.a. centrées, de même variance (finie), non corrélées, indépendantes des  $\varepsilon_i$ ,  
 $i = 1, \dots, n$ . Le paramètre  $\rho$  représente ainsi l'auto-corrélation :  
 $\text{Cor}(\varepsilon_i, \varepsilon_{i-1}) = \rho, \quad i = 2, \dots, n.$

La statistique de Durbin-Watson est définie par

$$S_{DW} := \frac{\sum_{i=2}^n (\hat{\varepsilon}_i - \hat{\varepsilon}_{i-1})^2}{\sum_{i=1}^n \hat{\varepsilon}_i^2}.$$

Elle prend ses valeurs entre 0 et 4. Le coefficient d'auto-corrélation  $\rho$  peut être estimé par  $1 - S_{DW}/2$ . L'hypothèse nulle est retenue si la statistique a une valeur proche de 2.

Pour réaliser ce test, sous R, on utilise la fonction “dwtest()”, du package “lmtest”, comme suit

```
library(lmtest)
modele.reg.multiple <- lm(formula = ozone ~ . , data = LAozoneData)
dwtest(modele.reg.multiple, alternative = c("two.sided"))

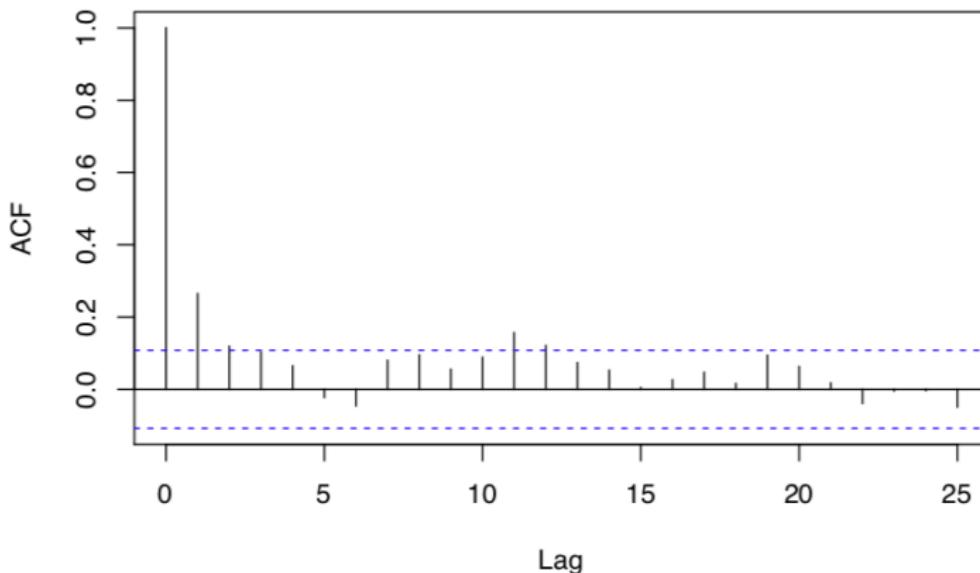
##
## Durbin-Watson test
##
## data: modele.reg.multiple
## DW = 1.4699, p-value = 3.837e-07
## alternative hypothesis: true autocorrelation is not 0
```

D'après ce test, les erreurs sont (fortement) corrélées.

Graphiquement, pour "vérifier" l'hypothèse de non-corrélation des erreurs, on peut tracer la fonction d'auto-corrélation des erreurs, à l'aide de la fonction "acf()" de R, comme suit

```
acf(modele.reg.multiple$residuals, main = "Autocorrélations des erreurs")
```

**Autocorrélations des erreurs**



## Test d'homoscédasticité des erreurs en régression : test de Breusch-Pagan

- Considérons le test de l'hypothèse nulle

$\mathcal{H}_0$  : “l'erreur  $\varepsilon$  est homoscédastique”

contre l'alternative

$\mathcal{H}_1$  : “l'erreur est hétéroscédastique”.

Pour tester l'hypothèse nulle ci-dessus (d'homoscédasticité), on peut utiliser, à partir des résidus, ou de leurs versions **studentisées**

$$\frac{\hat{\varepsilon}_i}{\hat{\sigma} \sqrt{1 - h_{i,i}}}, \quad i = 1, \dots, n,$$

$h_{i,i}$  étant le  $i$ -ème élément diagonal de la matrice  $\mathbb{X}(\mathbb{X}^\top \mathbb{X})^{-1}\mathbb{X}^\top$ , la statistique de **Breusch-Pagan** qui suit, asymptotiquement, sous  $\mathcal{H}_0$ , une loi du  $\chi_p^2$ ,  $p + 1$  étant le nombre de paramètres à estimer dans le modèle de RLM.

Cela peut se faire, sous R, à l'aide de la fonction “bptest()” du package “lmtest”, comme suit (à partir des résidus non studentisées)

```
modele.reg.multiple <- lm(formula = ozone ~ . , data = LAozoneData)
bptest(modele.reg.multiple, studentize = FALSE)

##
## Breusch-Pagan test
##
## data: modele.reg.multiple
## BP = 52.284, df = 9, p-value = 3.994e-08
```

où la version studentisée (i.e. avec des résidus studentisées)

```
modele.reg.multiple <- lm(formula = ozone ~ . , data = LAozoneData)
bptest(modele.reg.multiple)

##
## studentized Breusch-Pagan test
##
## data: modele.reg.multiple
## BP = 53.028, df = 9, p-value = 2.887e-08
```

La P-value est proche de zéro, on rejette alors l'hypothèse nulle d'homoscédasticité.

### Test de normalité des erreurs en régression

Considérons le test d'adéquation de l'hypothèse nulle  $\mathcal{H}_0$  : "l'erreur  $\varepsilon$  est normale" contre l'alternative  $\mathcal{H}_1$  : "l'erreur  $\varepsilon$  n'est pas normale". Les observations (résidus)  $\hat{\varepsilon}_1, \dots, \hat{\varepsilon}_n$ , ou leurs versions studentisées

$$\frac{\hat{\varepsilon}_i}{\hat{\sigma} \sqrt{1 - h_{i,i}}}, \quad i = 1, \dots, n,$$

$h_{i,i}$  étant le  $i$ -ème élément diagonal de la matrice  $\mathbb{X}(\mathbb{X}^\top \mathbb{X})^{-1}\mathbb{X}^\top$ , sont ensuite utilisées pour réaliser le test de normalité. Le test de **Shapiro-Wilk** est souvent utilisé ; sous R, on utilise la fonction 'shapiro.test()' :

## Test de normalité de l'erreur, à l'aide des résidus

```
shapiro.test(residuals(modele.reg.multiple))

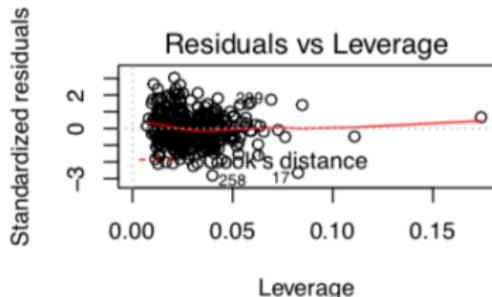
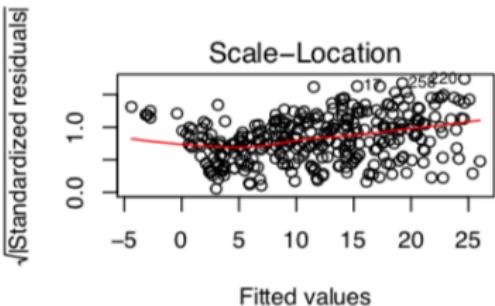
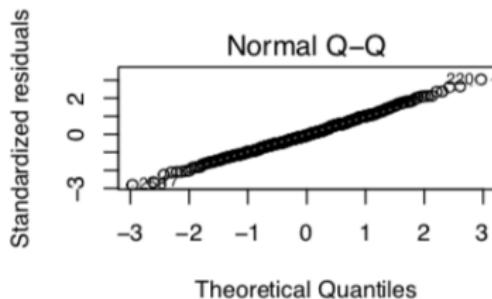
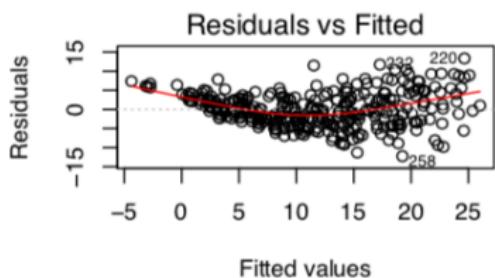
##
##  Shapiro-Wilk normality test
##
## data: residuals(modele.reg.multiple)
## W = 0.99697, p-value = 0.797
```

## Test de normalité de l'erreur, à l'aide des résidus studentisés

```
shapiro.test(rstudent(modele.reg.multiple))

##
##  Shapiro-Wilk normality test
##
## data: rstudent(modele.reg.multiple)
## W = 0.99705, p-value = 0.8138
```

Les graphiques suivants peuvent donner des indications quant à la validité ou non des hypothèses du modèle RL



Dans chacune des figures précédentes, on a présenté les 3 observations les plus extrêmes selon certains critères ; les critères utilisés, pour réaliser les graphiques précédents, peuvent être obtenues, à l'aide de la fonction 'augment()', comme suit

```
library(broom)
model.reg.multiple.metriques <- augment(modele.reg.multiple)
head(model.reg.multiple.metriques[,11:17])

## # A tibble: 6 x 7
##   .fitted    .se.fit   .resid   .hat   .sigma   .cooksdi   .std.resid
##     <dbl>      <dbl>    <dbl>   <dbl>     <dbl>       <dbl>
## 1    2.13      0.800   0.867  0.0324    4.45  0.000132    0.198
## 2    7.22      0.820  -2.22   0.0341    4.45  0.000915   -0.509
## 3   10.8       0.756  -5.75   0.0290    4.44  0.00516     -1.31
## 4    6.65      1.08   -0.646  0.0588    4.45  0.000140   -0.150
## 5    4.60      0.992  -0.604  0.0498    4.45  0.000102   -0.140
## 6    8.90      0.837  -4.90   0.0355    4.44  0.00464    -1.12
```

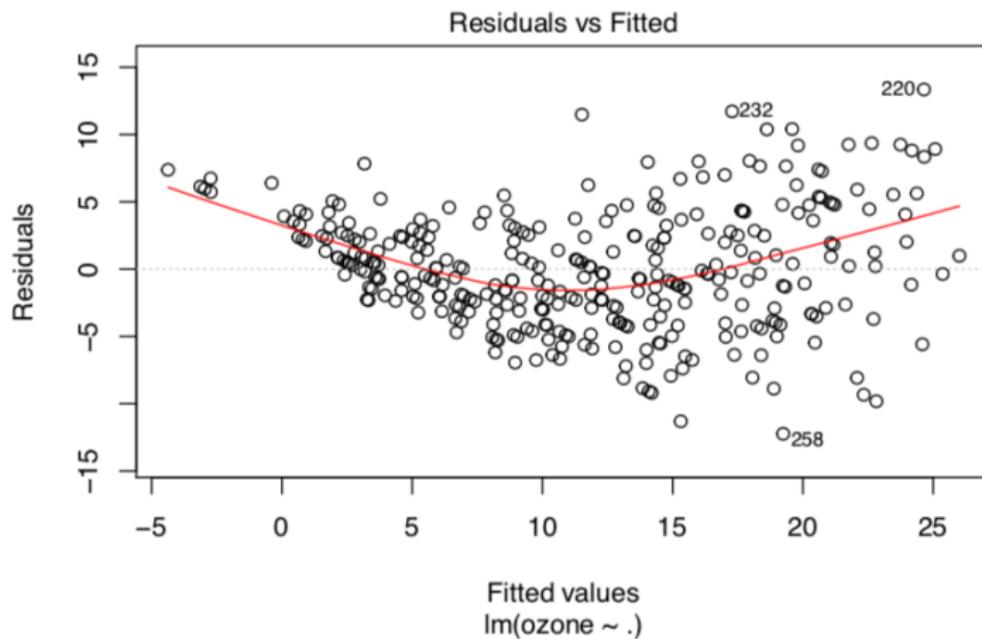
On utilise essentiellement

- **.fitted** : les valeurs ajustées ;
- **.resid** : les résidus ;
- **.hat** : les valeurs “hat” (les statistiques du levier) qui sont utilisées pour détecter les points leviers extrêmes (les valeurs extrêmes des prédicteurs) ;
- **.std.resid** : les résidus studentisés, qui peuvent être utilisés pour détecter les outliers (valeurs extrêmes de la variable réponse) ;
- **.cooksdi** : la distance de Cook qui est utilisée pour détecter les observations **excessivement influentes** (qui peuvent être des outliers ou points leviers extrêmes).

Dans ce qui suit, on montre comment utiliser ces figures pour vérifier les hypothèses du modèle de RLM, détecter des outliers/points levier extrêmes/points excessivement influents :

## Hypothèse de linéarité entre variable réponse et variables explicatives

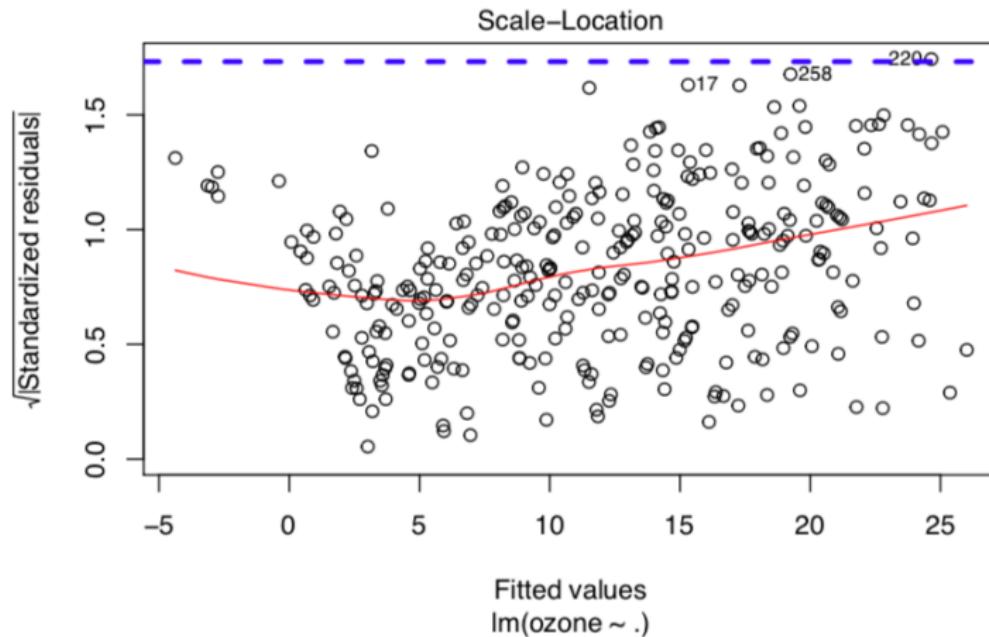
```
plot(modele.reg.multiple,1)
```



- La courbe rouge, dans la figure précédente, est obtenue par application d'une régression linéaire "local" des résidus en fonction des valeurs ajustées ; on peut utiliser la fonction 'loess()' de R ;
- Interprétation : une courbe horizontale indique une relation linéaire entre variable réponse et variables explicatives ; ce n'est pas le cas de la figure précédente : le lien, entre la variable réponse "ozone" et les variables explicatives, n'est pas linéaire, ...
- Les trois points ayant les plus grandes valeurs, en termes des résidus, sont représentés avec leurs indices ;
- On peut changer le nombre de points "extrêmes" à afficher en utilisant l'argument *id.n* dans la fonction "plot" ci-dessus.

## Hypothèse d'homogénéité des variances (homoscédasticité)

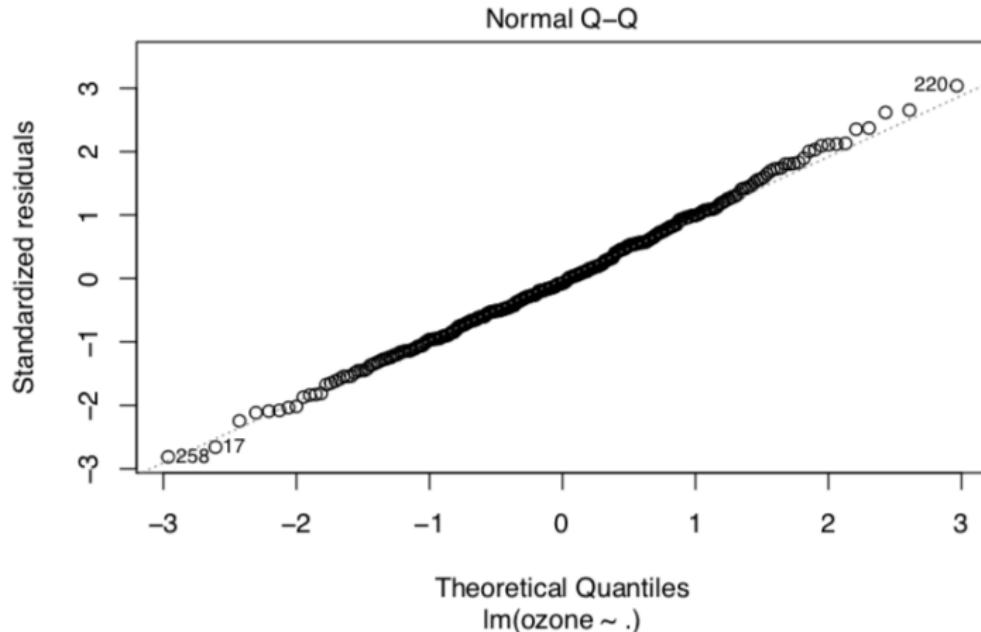
```
plot(modele.reg.multiple,3)
abline(h = sqrt(3), col = "blue", lty = 2, lwd = 3)
```



- **Interprétation** : une courbe horizontale, et des points répartis de manière homogène de part et d'autre le long des plages des valeurs prédites, indiquent une homogénéité des erreurs ; ce n'est pas le cas ici. (une transformation  $\log(\cdot)$  ou  $\sqrt{\cdot}$ , des valeurs de la variable réponse, pourrait remédier à ce problème, ...)
- Les trois points ayant les plus grandes valeurs, en termes de  $\sqrt{|r\acute{e}siduals studentisées|}$ , sont représentés avec leurs indices ;
- On peut changer le nombre de points “extrêmes” à afficher en utilisant l'argument *id.n* dans la fonction “plot” ci-dessus.

## Hypothèse de normalité de l'erreur

```
plot(modele.reg.multiple,2)
```



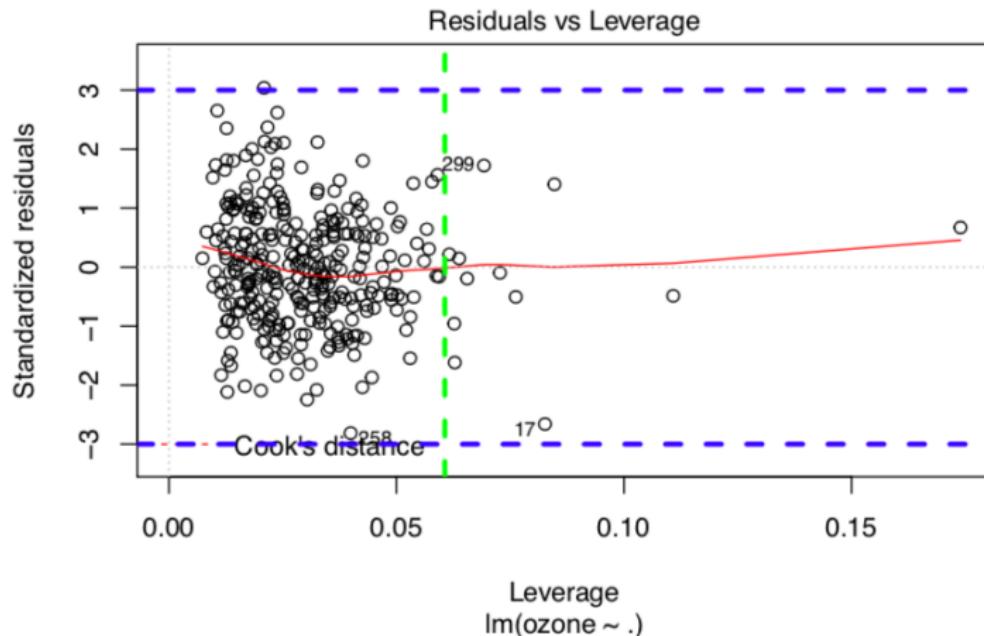
- **Interprétation** : Si tous les points se situent approximativement le long de la droite, cela indique que le terme d'erreur suit une loi normale. C'est le cas ici.

## Outliers et points leviers extrêmes

- Sont considérées outliers, les observations dont les résidus studentisés **sont supérieurs à 3** en valeur absolue (c.f. James et al. 2014) ;
- Une observation, dont la valeur de la “statistique du levier (ou hat.value)” est élevée, a des valeurs extrêmes en termes des variables explicatives. Une hat.value (ou statistique du levier), **supérieure à  $2(p + 1)/n$** , indique un point levier extrême (c.f. P. Bruce and A. Bruce 2017) ;
- Le graphique suivant peut être utilisé pour détecter les outliers et les points leviers extrêmes :

## Outliers et points leviers extrêmes

```
plot(modele.reg.multiple,5)
abline(h = c(-3,3), v = 2*(9 + 1)/330, col = c("blue","blue","green"), lty = c(2,2,2),
lwd = c(3,3,3))
```



## Interprétation :

- Tous les points ayant un “leverage” supérieur à  $2 * (p + 1) / n = 2 * (9 + 1) / 330 = 0.06$  sont considérés comme des points leviers extrêmes ;
- Tous les points ayant un résidu studentisé supérieur à 3 en valeur absolue sont considérés comme outliers.

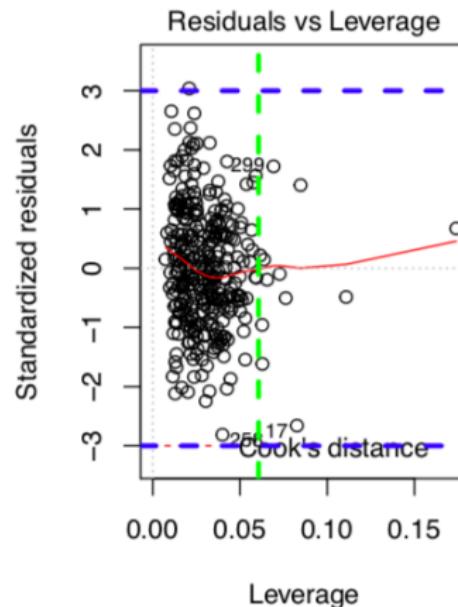
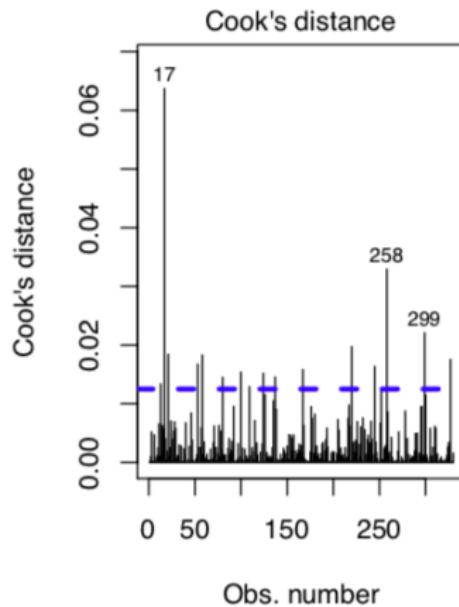
## Points influents

- La distance de Cook est souvent utilisée pour mesurer l'influence d'une observation dans le modèle ;
- Une observation est considérée comme **excessivement influente** si sa distance de Cook est **supérieure à  $4/(n - p - 1)$** .

On peut utiliser le graphique suivant

## Points influents

```
par(mfrow = c(1,2))
plot(modele.reg.multiple,4) #Distance de Cook
abline(h = 4/(330-9-1), col = c("blue"), lty = 2, lwd = 3)
plot(modele.reg.multiple,5) #Résidus versus Leverage
abline(h = c(-3,3), v = 2*(9 + 1)/330, col = c("blue","blue","green"), lty = c(2,2,2),
lwd = c(3,3,3))
```



## Équation de l'analyse de variance

On a d'après le théorème de Pythagore

$$\begin{aligned}\|\mathbf{Y} - \bar{\mathbf{Y}}\mathbf{1}\|^2 &= \|\hat{\mathbf{Y}} - \bar{\mathbf{Y}}\mathbf{1}\|^2 + \|\hat{\boldsymbol{\epsilon}}\|^2 \\ TSS &= RSS + SSE.\end{aligned}$$

(somme totale des carrés = somme des carrés de la régression + somme des carrés résiduels)

(total sum of squares = regression sum of squares + sum of squared errors)

## Coefficient de détermination $R^2$

$$R^2 := \frac{\|\hat{\mathbf{Y}} - \bar{\mathbf{Y}}\mathbf{1}\|^2}{\|\mathbf{Y} - \bar{\mathbf{Y}}\mathbf{1}\|^2} =: \frac{RSS}{TSS}.$$

- $0 \leq R^2 \leq 1$ ;
- Si  $R^2 = 1$ , la variabilité de la variable réponse est entièrement expliquée par le modèle ;
- Si  $R^2 = 0$ , toute la variabilité se trouve dans le bruit.

## Test du modèle global

- Le modèle :

$$Y_i = w_0 + w_1 X_{i,1} + \cdots + w_p X_{i,p} + \varepsilon_i, \quad i = 1, \dots, n,$$

avec  $\varepsilon_i, i = 1, \dots, n$ , i.i.d. de loi  $\mathcal{N}(0, \sigma^2)$ ,  
 (conditionnellement aux régresseurs).

- On veut tester

$$\mathcal{H}_0 : w_1 = \cdots = w_p = 0 \text{ contre } \mathcal{H}_1 : \exists j \in \{1, \dots, p\} \text{ t.q. } w_j \neq 0.$$

- Sous  $\mathcal{H}_0$ , la **statistique de Fisher**

$$F := \frac{R^2}{1 - R^2} \frac{n - p - 1}{p} = \frac{\|\bar{\mathbf{Y}}\mathbf{1} - \hat{\mathbf{Y}}\|^2 / p}{\|\mathbf{Y} - \hat{\mathbf{Y}}\|^2 / (n - p - 1)}$$

suit  $\mathcal{F}_{(p, n-p-1)}$ , la loi de Fisher à  $p$  et  $n - p - 1$  ddl.

- On rejette  $\mathcal{H}_0$  si  $F_{obs} > F_{(p,n-p-1)}(1 - \alpha)$ , où  $F_{(p,n-p-1)}(1 - \alpha)$  est le quantile d'ordre  $(1 - \alpha)$  de la loi  $\mathcal{F}_{(p,n-p-1)}$  ;
- La P-value associée est donc donnée par

$$P\text{-Value} = \mathbb{P}(Z > F_{obs}),$$

où  $Z$  est une v.a suivant la loi  $\mathcal{F}_{(p,n-p-1)}$ .

## Test entre modèles emboîtés

- On veut tester le modèle “réduit”  $\mathcal{M}_0$

$$Y_i = w_0 + w_{q+1}X_{i,q+1} + \cdots + w_pX_{i,p} + \varepsilon_i.$$

à “l’intérieur du modèle” plus large  $\mathcal{M}$

$$Y_i = w_0 + w_1X_{i,1} + \cdots + w_pX_{i,p} + \varepsilon_i;$$

- Cela revient à tester (à l’intérieur du modèle  $\mathcal{M}$ ) l’hypothèse nulle

$$\mathcal{H}_0 : w_1 = \cdots = w_q = 0$$

contre l’alternative

$$\mathcal{H}_1 : \exists j \in \{1, \dots, q\} \text{ t.q. } w_j \neq 0.$$

- Pour réaliser le test précédent, on utilise la statistique de Fisher suivante

$$F := \frac{\|\hat{\mathbf{Y}}_0 - \hat{\mathbf{Y}}\|^2/q}{\|\mathbf{Y} - \hat{\mathbf{Y}}\|^2/(n-p-1)}$$

qui suit la loi  $\mathcal{F}_{q,n-p-1}$ , sous  $\mathcal{H}_0$ .

( $\hat{\mathbf{Y}}_0$  désigne le vecteur des valeurs ajustées du modèle réduit).

- On rejette  $\mathcal{H}_0$  si  $F_{obs} > F_{(q,n-p-1)}(1-\alpha)$ .
- La  $P$ -value est donc définie par

$$P\text{-value} := \mathbb{P}(Z > F_{obs}),$$

où  $Z$  est une v.a. suivant la loi  $\mathcal{F}_{(q,n-p-1)}$ .

- Ce test peut se faire sous R à l'aide de la fonction 'anova()' qu'on applique à des modèles linéaires calculés avec la fonction 'lm()' :

## Retour à l'exemple 1 : test du modèle global

```
modele.complet <- lm(formula = ozone ~ . , data = LAozoneData)
modele.trivial <- lm(formula = ozone ~ 1, data = LAozoneData)
anova(modele.trivial, modele.complet)

## Analysis of Variance Table
##
## Model 1: ozone ~ 1
## Model 2: ozone ~ vh + wind + humidity + temp + ibh + dpg + ibt + vis +
##           doy
##   Res.Df   RSS Df Sum of Sq    F    Pr(>F)
## 1    329 21115.4
## 2    320  6311.2  9     14804 83.403 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Retour à l'exemple 1 : test entre deux modèles emboîtés

```
modele.complet <- lm(formula = ozone ~ . , data = LAozoneData)
modele.reduit <- lm(formula = ozone ~ . -dpg-vis, data = LAozoneData)
anova(modele.reduit,modele.complet)

## Analysis of Variance Table
##
## Model 1: ozone ~ (vh + wind + humidity + temp + ibh + dpg + ibt + vis +
##      doy) - dpg - vis
## Model 2: ozone ~ vh + wind + humidity + temp + ibh + dpg + ibt + vis +
##      doy
##   Res.Df   RSS Df Sum of Sq    F  Pr(>F)
## 1     322 6404.7
## 2     320 6311.2  2    93.508 2.3706 0.09506 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Tests simples, relatifs aux paramètres du modèle de RLM : on utilise la statistique de Student ou Fisher ?

Ici les résultats du test de Student :

```
modele.complet <- lm(formula = ozone ~ . , data = LAozoneData)
summary(modele.complet)

##
## Call:
## lm(formula = ozone ~ . , data = LAozoneData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.2407  -2.8832  -0.3353   2.7409  13.3523
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 18.3792938 29.5045242  0.623  0.53377
## vh          -0.0051340  0.0053950 -0.952  0.34200
## wind         -0.0198304  0.1238829 -0.160  0.87292
## humidity     0.0804923  0.0188345  4.274 2.54e-05 ***
## temp          0.2743349  0.0497361  5.516 7.17e-08 ***
## ibh          -0.0002497  0.0002950 -0.846  0.39798
## dpg          -0.0036968  0.0112925 -0.327  0.74360
## ibt          0.0292640  0.0136115  2.150  0.03231 *
## vis          -0.0080742  0.0037565 -2.149  0.03235 *
## doy          -0.0088490  0.0027199 -3.253  0.00126 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.441 on 320 degrees of freedom
## Multiple R-squared:  0.7011, Adjusted R-squared:  0.6927
## F-statistic:  83.4 on 9 and 320 DF,  p-value: < 2.2e-16
```

## Voici les résultats du test de Fisher

```
modele.complet <- lm(formula = ozone ~ . , data = LAozoneData)
anova(modele.complet)

## Analysis of Variance Table

## Response: ozone
##             Df Sum Sq Mean Sq F value    Pr(>F)
## vh          1 7788.8 7788.8 394.9172 < 2.2e-16 ***
## wind        1  406.5  406.5 20.6130 7.969e-06 ***
## humidity    1 3100.3 3100.3 157.1959 < 2.2e-16 ***
## temp         1 2415.2 2415.2 122.4578 < 2.2e-16 ***
## ibh          1   735.0   735.0 37.2678 2.978e-09 ***
## dpg          1    12.5    12.5  0.6313  0.427464
## ibt          1    76.4    76.4  3.8745  0.049888 *
## vis          1    60.8    60.8  3.0810  0.080167 .
## doy          1   208.8   208.8 10.5851  0.001262 **
## Residuals 320 6311.2   19.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Remarque

- Il est préférable d'utiliser le test de Fisher (au lieu du test de Student), en particulier dans le cas de régression linéaire multiple (avec variables explicatives dépendantes).

## Taille d'un modèle et précision

- Lorsque la taille du modèle est petite (ou nombre petit de variables explicatives) :
  - variance faible, biais élevé ;
  - erreur théorique de prévision élevée ;
  - erreur empirique (d'ajustement) élevée ;
- Lorsque la taille du modèle est grande (ou nombre élevé de variables explicatives) :
  - variance élevée, biais faible ;
  - erreur théorique de prévision élevée ;
  - erreur empirique (d'ajustement) très faible (**problème de sur-ajustement**, appelé aussi **problème de sur-apprentissage !**) ;
- D'où la nécessité de développer des procédures de **sélection de modèles (de variables)**.

# Algorithmes de sélection de modèles

## Le cadre

- On considère un modèle de RLM

$$Y = w_0 + w_1 X_1 + \dots + w_p X_p + \varepsilon$$

- On dispose d'un échantillon  $(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)$  du vecteur  $(\mathbf{X}, Y) =: (X_1, \dots, X_p, Y) \in \mathbb{R}^p \times \mathbb{R}$ .
- L'objectif est d'obtenir le sous-ensemble de variables explicatives qui conduit au “meilleur” modèle RLM au sens d'un **critère** donné.
- Avec  $p$  variables explicatives candidates,  $X_1, \dots, X_p$ , on peut construire  $2^p - 1$  modèles de régression linéaires différents, d'ordre 1 et sans interactions, (les modèles à une variable, à deux variables, ...).

## Approche exhaustive

- Construire les  $2^P - 1$  modèles ;
- Choisir celui qui optimise un **critère** donné.

## Exemples de critères de sélection

- (1) L'Akaike Information Criterion (**AIC**), d'un modèle de RLM, constitué de  $k$  variables explicatives, est défini par

$$AIC = -2 \mathcal{L}_n(\hat{\mathbf{w}}, \tilde{\sigma}^2) + 2(k+2),$$

où  $\mathcal{L}_n(\hat{\mathbf{w}}, \tilde{\sigma}^2)$  est la log-vraisemblance du modèle, définie ci-dessus, sous les hypothèses d'homoscédasticité et de normalité des erreurs ; c.f. (3) ;

- (2) Le Bayesian Information Criterion (**BIC**) :

$$BIC = -2 \mathcal{L}_n(\hat{\mathbf{w}}, \tilde{\sigma}^2) + \log(n)(k+2);$$

- (3) **R<sup>2</sup> ajusté** :  $R_a^2 := 1 - \frac{n-1}{n-k-1} (1 - R^2)$  ;

- (4) Le  $C_p$  de Mallow d'un modèle de régression utilisant  $k$  variables explicatives ( $1 \leq k \leq p$ ) est donné par :

$$C_p := \frac{1}{n} \left( \|\mathbf{Y} - \hat{\mathbf{Y}}_0 \mathbf{1}\|^2 + 2(1+k)\hat{\sigma}^2 \right),$$

où  $\hat{\mathbf{Y}}_0$  est le vecteur des valeurs ajustées selon le modèle utilisant les  $k$  variables explicatives ;

- (5) Le critère  $F$  de Fisher : Notons  $\hat{\mathbf{Y}}$  le vecteur des valeurs ajustées selon le modèle de RLM complet à  $p$  variables explicatives, et  $\hat{\mathbf{Y}}_0$  le vecteur des valeurs ajustées selon le modèle réduit à  $p - q$  variables explicatives ( $1 \leq q < p$ ). Le critère  $F$  du modèle réduit est défini par

$$F := \frac{\|\hat{\mathbf{Y}}_0 - \hat{\mathbf{Y}}\|^2/q}{\|\mathbf{Y} - \hat{\mathbf{Y}}\|^2/(n-p-1)}.$$

Sous R, la méthode précédente peut se faire à l'aide de la fonction 'regsubsets()' du package 'leaps', comme suit

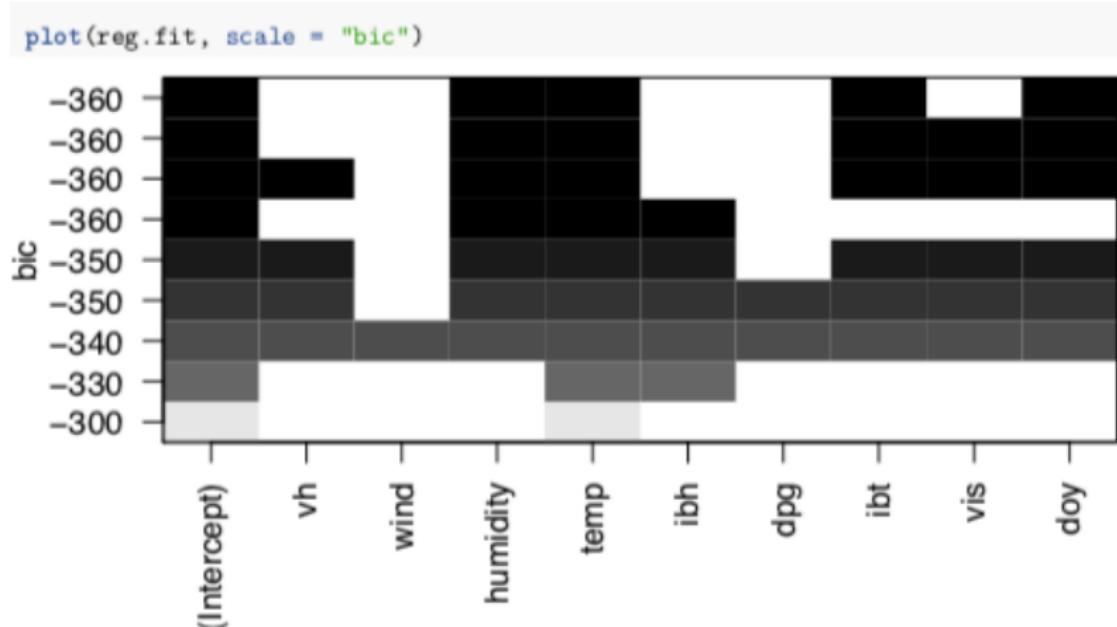
```
library(leaps)
reg.fit <- regsubsets(ozone ~ ., data = LAozoneData, nvmax = NULL, method = "exhaustive")
summary(reg.fit)

## Subset selection object
## Call: regsubsets.formula(ozone ~ ., data = LAozoneData, nvmax = NULL,
##   method = "exhaustive")
## 9 Variables (and intercept)
##          Forced in Forced out
## vh      FALSE    FALSE
## wind    FALSE    FALSE
## humidity FALSE   FALSE
## temp    FALSE   FALSE
## ibh     FALSE   FALSE
## dpg     FALSE   FALSE
## ibt     FALSE   FALSE
## vis     FALSE   FALSE
## doy     FALSE   FALSE
## 1 subsets of each size up to 9
## Selection Algorithm: exhaustive
##           vh wind humidity temp ibh dpg ibt vis doy
## 1 ( 1 ) " " " " " * " " " " " "
## 2 ( 1 ) " " " " " * " * " " " " "
## 3 ( 1 ) " " " " * " * " * " " " "
## 4 ( 1 ) " " " " * " * " * " * " "
## 5 ( 1 ) " " " " * " * " * " * " "
## 6 ( 1 ) * " " " * " * " * " * " "
## 7 ( 1 ) * " " " * " * " * " * " "
## 8 ( 1 ) * " " " * " * " * " * " "
## 9 ( 1 ) * " " * " * " * " * " * "
```

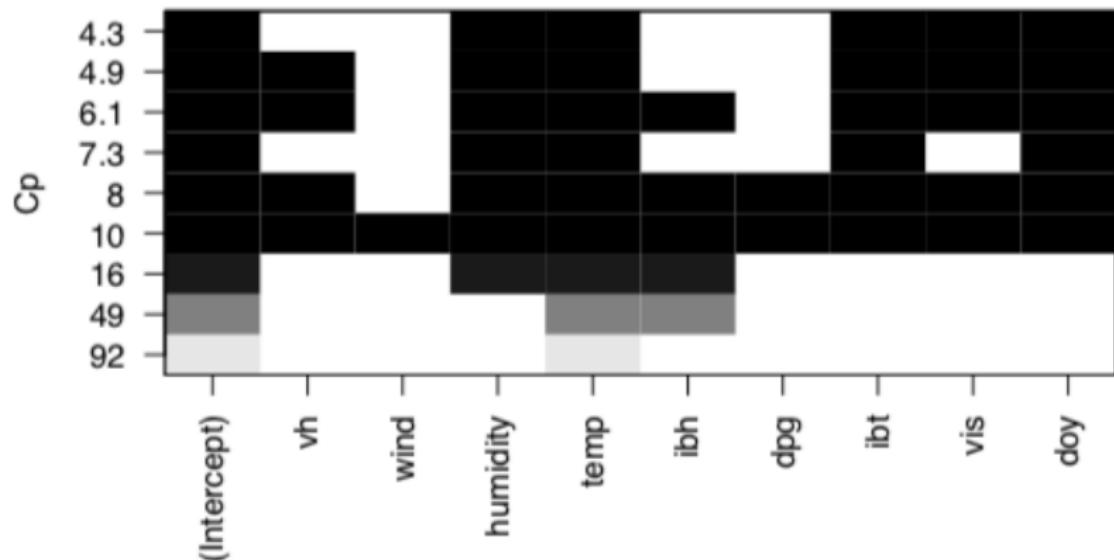
On donne ici les valeurs des critères  $R^2$ ,  $R^2$  ajusté, BIC et  $C_p$ , de chacun des modèles optimaux (de dimension 2, 3, ...,  $p$ ) :

```
summary(reg.fit)$rsq
## [1] 0.6094969 0.6516176 0.6839717 0.6943308 0.6989193 0.7002759 0.7009729
## [8] 0.7010849 0.7011088
summary(reg.fit)$adjr2
## [1] 0.6083064 0.6494868 0.6810634 0.6905687 0.6942730 0.6947083 0.6944723
## [8] 0.6936353 0.6927025
summary(reg.fit)$bic
## [1] -298.7072 -330.5727 -356.9384 -362.1377 -361.3298 -357.0210 -351.9902
## [8] -346.3147 -340.5420
summary(reg.fit)$cp
## [1] 92.081837 48.986427 16.347403 7.256658 4.344125 4.891726 6.145519
## [8] 8.025624 10.000000
```

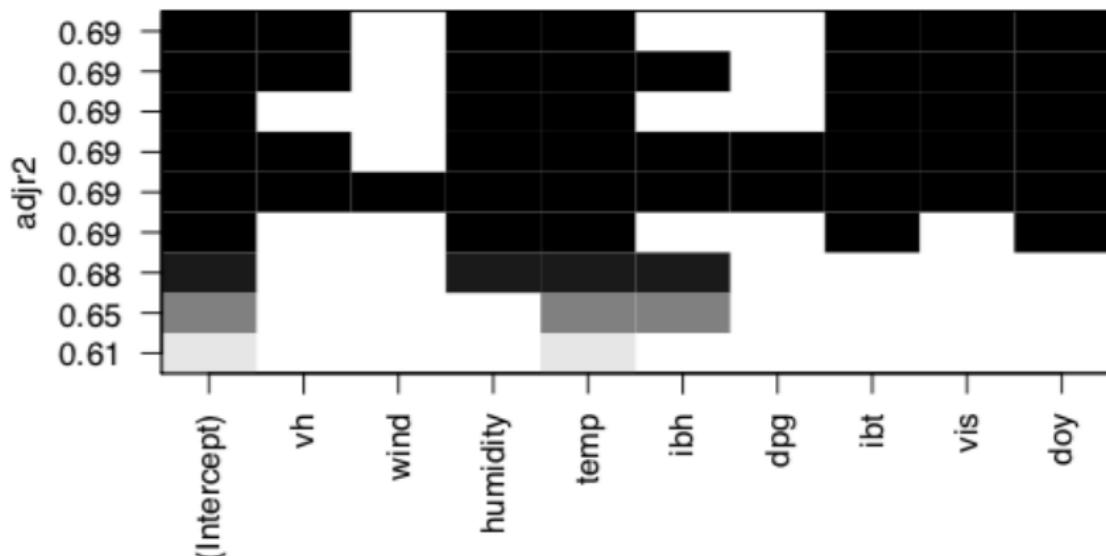
Les résultats de sélection, selon un critère donné, peuvent être représentés graphiquement comme suit



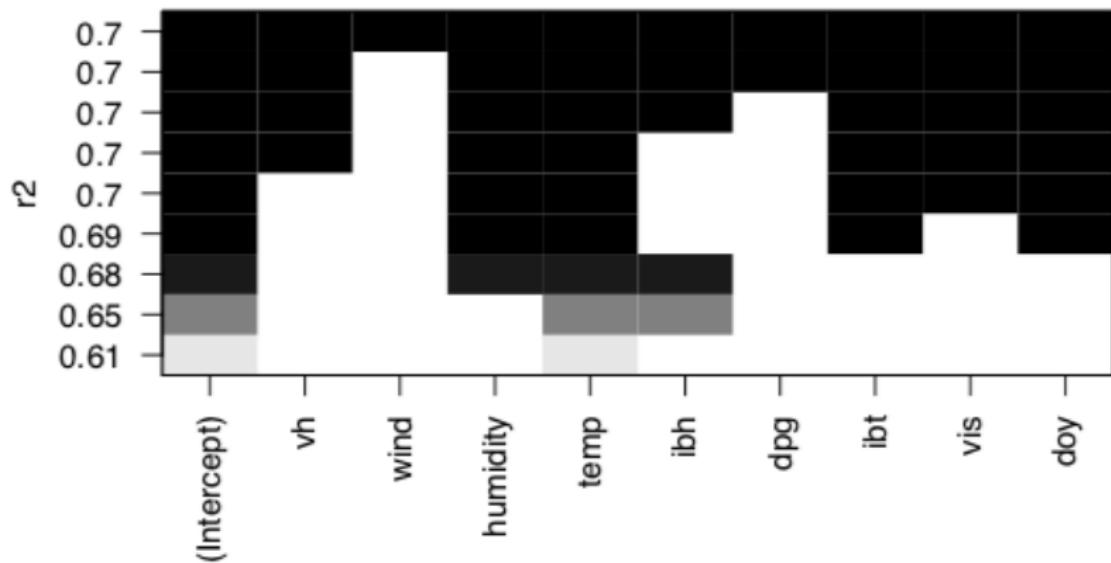
```
plot(reg.fit, scale = "Cp")
```



```
plot(reg.fit, scale = "adjr2")
```



```
plot(reg.fit, scale = "r2")
```



- Remarquer que le critère  $R^2$  sélectionne automatiquement le plus grand modèle (il n'est donc pas utilisé pour la selection de modèles).

On peut également utiliser la fonction 'glmulti()' du package 'glmulti'

```
library(glmulti)
select.aic <- glmulti(ozone ~ ., data = LAozoneData, level = 1,
                      method = "h", fitfunction = lm, crit = "aic", plotty = FALSE)

aic.best.model <- summary(select.aic)$bestmodel
anova(lm(aic.best.model, data = LAozoneData))

## Analysis of Variance Table
##
## Response: ozone
##             Df Sum Sq Mean Sq F value    Pr(>F)
## humidity     1 4261.1 4261.1 217.1641 < 2.2e-16 ***
## temp         1 9412.1 9412.1 479.6786 < 2.2e-16 ***
## ibt          1  693.0  693.0  35.3193 7.229e-09 ***
## vis          1   48.1   48.1   2.4511   0.1184
## doy          1   343.6   343.6  17.5104 3.684e-05 ***
## Residuals 324 6357.4    19.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
select.bic <- glmulti(ozone ~., data = LAozoneData, level = 1,
                      method = "h", fitfunction = lm, crit = "bic", plotty = FALSE)

bic.best.model <- summary(select.bic)$bestmodel
anova(lm(bic.best.model, data = LAozoneData))

## Analysis of Variance Table
##
## Response: ozone
##             Df Sum Sq Mean Sq F value    Pr(>F)
## humidity     1 4261.1  4261.1 214.564 < 2.2e-16 ***
## temp         1 9412.1  9412.1 473.936 < 2.2e-16 ***
## ibt          1   693.0   693.0  34.897 8.769e-09 ***
## doy          1   294.8   294.8  14.844  0.0001407 ***
## Residuals 325 6454.3     19.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- L'approche exhaustive permet de comparer tous les modèles ; l'inconvénient est que le temps de calcul devient très important si le nombre de variables est grand ;
- Lorsque le nombre de variables est grand, on privilégie souvent les méthodes “**pas à pas**” qui consistent à construire les modèles de façon “réursive”, en ajoutant/supprimant une variable explicative à chaque étape.

## Méthode ascendante (forward selection, version 1)

1. Construire  $\mathcal{M}_0$  le modèle trivial (avec uniquement l'intercept) ;
2. Pour  $k = 0, \dots, p - 1$  :
  - 2.1. Construire les  $p - k$  modèles consistant à ajouter une variable dans  $\mathcal{M}_k$  ;
  - 2.2. Choisir, parmi ces  $p - k$  modèles, celui qui optimise un critère donné ;
3. Choisir, parmi  $\mathcal{M}_1, \dots, \mathcal{M}_p$ , le meilleur modèle au sens du critère considéré.

## Méthode descendante (backward elimination, version 1)

1. Construire  $\mathcal{M}_p$  le modèle complet (avec les  $p$  variables) ;
2. Pour  $k = p, \dots, 1$  :
  - 2.1. Construire les  $k$  modèles consistant à supprimer une variable dans  $\mathcal{M}_k$  ;
  - 2.2. Choisir, parmi ces  $k$  modèles, celui qui optimise un critère donné ;
3. Choisir, parmi  $\mathcal{M}_1, \dots, \mathcal{M}_p$ , le meilleur modèle au sens du critère considéré.

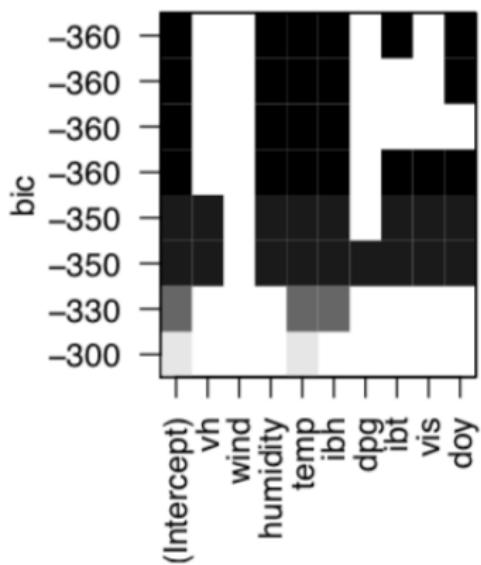
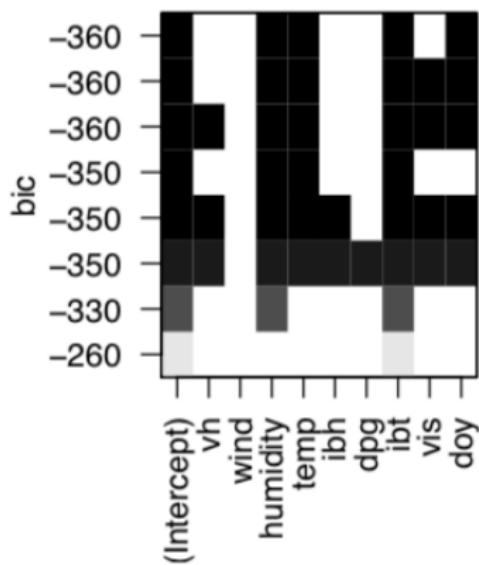
## Application sous R

Pour appliquer ces deux dernières méthodes, sous R, on utilise la fonction 'regsubsets()', du package 'leaps', en spécifiant l'argument `method = "forward"` ou `method = "backward"`, comme suit

```
select.mod.for <- regsubsets(ozone ~ ., data = LAozoneData, method = "forward")
select.mod.bac <- regsubsets(ozone ~ ., data = LAozoneData, method = "backward")
```

Voici les résultats obtenus pour le critère BIC :

```
par(mfrow = c(1, 2))
plot(select.mod.for, scale = "bic", main = "Forward selection")
plot(select.mod.bac, scale = "bic", main = "Backward elimination")
```

**Forward selection****Backward elimination**

Les quatre méthodes suivantes peuvent être appliquées, sous R, à l'aide de la fonction 'step()' en spécifiant les arguments 'direction = "forward"', 'direction = "backward"' ou 'direction = "both"'. Le critère AIC correspond à l'argument ' $k = 2$ ', le critère BIC correspond à ' $k = \log n$ ',  $n$  étant le nombre d'observations. Le critère de Fisher, pour la sélection de modèles, peut être utilisé à l'aide de l'argument 'test = "F"'.

### Méthode ascendante (forward selection, version 2, pour un critère à minimiser)

1. Modèle sans variables ;
2. Insertion de la variable qui diminue le plus le critère ;
3. Insertion de la deuxième variable qui diminue le plus le critère,  
... arrêt quand on ne diminue plus le critère.

```
modele.trivial <- lm(ozone ~ 1, data = LAozoneData)
modele.complet <- lm(ozone ~ ., data = LAozoneData)

res.select.AIC.for <- step(modele.trivial,
                           scope = list(lower = modele.trivial, upper = modele.complet),
                           data = LAozoneData, direction = "forward", k = 2)

n <- nrow(LAozoneData)
res.select.BIC.for <- step(modele.trivial,
                           scope = list(lower = modele.trivial, upper = modele.complet),
                           data = LAozoneData, direction = "forward", k = log(n))

res.select.F.for <- step(modele.trivial,
                           scope = list(lower = modele.trivial, upper = modele.complet),
                           data = LAozoneData, direction = "forward", test = "F")
```

## Méthode descendante (backward elimination, version 2, pour un critère à minimiser)

1. Modèle complet ;
2. Enlever la variable qui diminue le plus le critère ;
3. Enlever la deuxième variable qui diminue le plus le critère, ... arrêt quand on ne diminue plus le critère.

```
modele.complet <- lm(ozone ~ ., data = LAozoneData)

res.select.AIC.bac <- step(modele.complet, data = LAozoneData,
                           direction = "backward", k = 2)

res.select.BIC.bac <- step(modele.complet, data = LAozoneData,
                           direction = "backward", k = log(n))

res.select.F.bac <- step(modele.complet, data = LAozoneData,
                           direction = "backward", test = "F")
```

## Méthode ascendante bidirectionnelle (bidirectional selection)

- Ascendante avec remise en cause à chaque étape des variables déjà incluses ;
- Permet d'exclure des variables qui redeviennent plus significatives compte tenu de celle qui vient d'être intégrée.

```
modele.trivial <- lm(ozone ~ 1, data = LAozoneData)
modele.complet <- lm(ozone ~ ., data = LAozoneData)
res.select.AIC.for.both <- step(modele.trivial, scope = list(lower = modele.trivial,
                                                               upper = modele.complet),
                                 data = LAozoneData, direction = "both", k = 2)

res.select.BIC.for.both <- step(modele.trivial, scope = list(lower = modele.trivial,
                                                               upper = modele.complet),
                                 data = LAozoneData, direction = "both", k = log(n))

res.select.F.for.both <- step(modele.trivial, scope = list(lower = modele.trivial,
                                                               upper = modele.complet),
                                 data = LAozoneData, direction = "both", test = "F")
```

## Méthode descendante bidirectionnelle (bidirectional elimination)

- Descendante avec remise en cause à chaque étape des variables déjà exclues ;
- Permet de réintégrer des variables qui redeviennent significatives compte tenu de celle qui vient d'être exclue.

```
modele.complet <- lm(ozone ~ ., data = LAozoneData)
res.select.AIC.bac.both <- step(modele.complet, data = LAozoneData,
                                 direction = "both", k = 2)

res.select.BIC.bac.both <- step(modele.complet, data = LAozoneData,
                                 direction = "both", k = log(n))

res.select.F.bac.both <- step(modele.complet, data = LAozoneData,
                               direction = "both", test = "F")
```

## Remarque

- Les méthodes pas à pas précédentes peuvent aboutir à des modèles différents et non optimaux.

## Sélection par algorithme génétique

- On l'utilise quand le nombre de variables devient de plus en plus important et qu'une recherche exhaustive est impossible et une recherche pas à pas peut mener à une solution qui n'est pas tout à fait optimale ;
- Cet algorithme est implémenté dans la commande 'glmulti()' du package 'glmulti' en spécifiant l'argument `method = "g"` ;
- Cette méthode est donc supposée trouver le meilleur modèle sans avoir besoin de calculer le critère à considérer sur tous les modèles possibles (recherche exhaustive).

```
select.mod.gen <- glmulti(ozone ~ ., data = LAozoneData, level = 1, method = "g",
                           fitfunction = lm, crit = 'aic', plotty = F)
```

```
aic.best.model <- summary(select.mod.gen)$bestmodel
aic.best.model
```

```
## [1] "ozone ~ 1 + humidity + temp + ibt + vis + doy"
```

```
select.mod.gen <- glmulti(ozone ~ ., data = LAozoneData, level = 1, method = "g",
                           fitfunction = lm, crit = 'bic', plotty = F)
```

```
bic.best.model <- summary(select.mod.gen)$bestmodel
bic.best.model
```

```
## [1] "ozone ~ 1 + humidity + temp + ibt + doy"
```

## Le bootstrap

- Le **bootstrap** est un outil performant pour évaluer l'incertitude d'un estimateur ou d'une méthode d'apprentissage. Par exemple, il peut être utilisé pour estimer les variances et les biais des estimateurs des paramètres d'un modèle de densité de probabilité ou d'un modèle de régression ;
- Nous allons illustrer cette méthode du bootstrap dans le cas simple de l'estimation paramétrique d'une densité de probabilité. Soit  $X$  une variable aléatoire réelle dont la densité de probabilité est supposée appartenir à un modèle paramétrique (une famille paramétrique de densités de proba.)  $\{x \in \mathbb{R} \mapsto f(x, \theta); \theta \in \Theta \subset \mathbb{R}\}$ . Notons  $\theta_v$  la vraie valeur inconnue du paramètre  $\theta$ . Soit  $D_n := \{X_1, \dots, X_n\}$  un échantillon i.i.d. de  $X$ . L'estimateur du MV  $\hat{\theta}$  de  $\theta_v$  est

$$\hat{\theta} := \arg \sup_{\theta \in \Theta} L(D_n, \theta) := \arg \sup_{\theta \in \Theta} \prod_{i=1}^n f(X_i, \theta).$$

- Il est naturel de vouloir évaluer l'efficacité de l'estimateur  $\hat{\theta}$ , i.e., estimer l'**erreur théorique**

$$\begin{aligned}\mathbb{E} \left\{ (\hat{\theta} - \theta_v)^2 \right\} &= \mathbb{E} \left\{ (\hat{\theta} - \mathbb{E}(\hat{\theta}))^2 \right\} + \left\{ \mathbb{E}(\hat{\theta}) - \theta_v \right\}^2 \\ &=: \text{Var}(\hat{\theta}) + \text{biais}(\hat{\theta})^2.\end{aligned}$$

La méthode du bootstrap, pour estimer  $\text{Var}(\hat{\theta})$ , consiste à générer un “grand” nombre  $R$  d’échantillons “**bootstrapés**” (chacun de taille  $n$ ) :  $D_r^* := \{X_{1,r}^*, \dots, X_{n,r}^*\}$ ,  $r = 1, \dots, R$ . Chacun des échantillons bootstrapé est obtenu en effectuant  $n$  tirages avec remise dans l’ensemble des observations  $\{X_1, \dots, X_n\}$ . On calcule alors  $R$  nouveaux estimateurs (**bootstrapés**),

$$\hat{\theta}_r^* := \arg \sup_{\theta \in \Theta} L(D_r^*, \theta) = \arg \sup_{\theta \in \Theta} \prod_{i=1}^n f(X_{i,r}^*, \theta), \quad r = 1, \dots, R,$$

chacun des estimateurs  $\widehat{\theta}_r^*$ ,  $r = 1 \dots, R$ , est calculé à partir de l'échantillon bootstrapé  $D_r^* := \{X_{1,r}^*, \dots, X_{n,r}^*\}$ . La variance  $\text{Var}(\widehat{\theta})$  est estimée alors par

$$\widehat{\text{Var}}(\widehat{\theta}) := \frac{1}{R-1} \sum_{r=1}^R \left( \widehat{\theta}_r^* - \frac{1}{R} \sum_{r=1}^R \widehat{\theta}_r^* \right)^2.$$

De même, le biais peut être estimé par

$$\widehat{\text{biais}}(\widehat{\theta}) := \left( \frac{1}{R} \sum_{r=1}^R \widehat{\theta}_r^* \right) - \widehat{\theta}.$$

On obtient alors l'estimation suivante de l'erreur quadratique de l'estimateur  $\widehat{\theta}$  :

$$\frac{1}{R-1} \sum_{r=1}^R \left( \widehat{\theta}_r^* - \frac{1}{R} \sum_{r=1}^R \widehat{\theta}_r^* \right)^2 + \left( \left( \frac{1}{R} \sum_{r=1}^R \widehat{\theta}_r^* \right) - \widehat{\theta} \right)^2.$$

## Comment sélectionner le meilleur modèle, et éviter le problème de sur-ajustement ?

Il faut estimer correctement l'erreur théorique de prévision de chaque modèle, et choisir le modèle ayant l'erreur estimée la plus faible. Les méthodes de validation croisée permettent d'évaluer efficacement cette erreur. On présente ici trois méthodes différentes :

- **Méthode 1** : (l'approche **apprentissage/validation**). Cette approche est très simple. Elle consiste à diviser de manière aléatoire l'ensemble d'observations en deux parties : un ensemble d'apprentissage ( $\mathcal{A}$ ) et un ensemble de validation ( $\mathcal{V}$ ). Le modèle est construit à partir des données d'apprentissage, et il est ensuite utilisé pour prédire la variable réponse des données de l'ensemble de validation. L'erreur théorique (de prévision) est alors estimée par l'écart moyen entre les valeurs prédites et les valeurs observées de la variable réponse des données de l'ensemble de validation :

$$\hat{\mathbf{w}} := \arg \min_{\mathbf{w}} \frac{1}{\text{Card}(\mathcal{A})} \sum_{i \in \mathcal{A}} \ell(Y_i, f(\mathbf{X}_i, \mathbf{w})),$$

et l'erreur théorique de prévision, que l'on note  $\hat{\mathcal{E}}$ , est estimée par

$$\hat{\mathcal{E}} := \frac{1}{\text{Card}(\mathcal{V})} \sum_{i \in \mathcal{V}} \ell(Y_i, f(\mathbf{X}_i, \hat{\mathbf{w}})). \quad (4)$$

Cette méthode est simple, mais elle a deux inconvénients :

- L'estimation de l'erreur théorique (de prévision) dépend de la partition, et elle peut varier beaucoup si on considère une partition différente ;
- Dans cette approche seulement une partie des données est utilisée pour ajuster le modèle, ce qui peut mener à une sur-estimation de l'erreur.

Les deux méthodes suivantes répondent à ces problèmes.

- Méthode 2 : (leave-one-out cross-validation (LOOCV)).

Comme la méthode précédente, la méthode LOOCV consiste à diviser l'ensemble des observations en deux parties.

Cependant, au lieu de créer deux parties de tailles comparables, une seule observation  $(\mathbf{X}_1, Y_1)$  est utilisée pour la validation et tout le reste est utilisé pour l'apprentissage (pour estimer le modèle). Donc le modèle est ajusté sur la base des  $(n - 1)$  observations restantes, et une prédiction  $\hat{Y}_1$  est calculée en fonction de  $\mathbf{X}_1$ . Plus précisément, si on note  $\hat{\mathbf{w}}^{(-1)}$  l'estimateur obtenu en utilisant toutes les observations sauf l'observation  $(\mathbf{X}_1, Y_1)$ , alors  $\hat{Y}_1 := f(\mathbf{X}_1, \hat{\mathbf{w}}^{(-1)})$ . Puisque l'observation  $(\mathbf{X}_1, Y_1)$  n'a pas été utilisée pour ajuster le modèle, alors  $e_1 := \ell(Y_1, \hat{Y}_1) = \ell(Y_1, f(\mathbf{X}_1, \hat{\mathbf{w}}^{(-1)}))$  est une estimation approximativement sans biais de l'erreur théorique (de prévision), mais elle n'est pas précise puisque elle est basée sur une seule observation.

On répète alors la procédure précédente en choisissant  $(\mathbf{X}_2, Y_2)$  pour la validation, et les observations restantes pour ajuster le modèle, et on calcule  $e_2 := \ell(Y_2, \hat{Y}_2) = \ell(Y_2, f(\mathbf{X}_2, \hat{\mathbf{w}}^{(-2)}))$ . En répétant la procédure  $n$  fois, on obtient  $n$  estimations,  $e_1, e_2, \dots, e_n$ , de l'erreur théorique (de prévision). Le LOOCV donne l'estimation suivante de l'erreur théorique de prévision

$$\widehat{\mathcal{E}} := \frac{1}{n} \sum_{i=1}^n e_i = \frac{1}{n} \sum_{i=1}^n \ell(Y_i, f(\mathbf{X}_i, \hat{\mathbf{w}}^{(-i)})). \quad (5)$$

Cette méthode a au moins deux avantages par rapport à la méthode 1 précédente (l'approche de l'ensemble de validation).

- En LOOCV, l'estimation de l'erreur de prévision est plus précise, car à chaque fois on utilise toutes les données (sauf une) pour estimer les paramètres du modèle, alors que la méthode 1 utilise seulement une partie des données pour l'estimation des paramètres du modèle ;

- Contrairement à la méthode apprentissage/validation -qui donne des résultats différents en changeant la répartition des données entre les deux ensembles apprentissage/validation- le LOOCV donne le même résultat s'il est appliqué plusieurs fois sur la même base de données.

Par ailleurs, le LOOCV est coûteux en temps de calcul en général, en particulier si  $n$  est grand, puisque le calcul des estimateurs des paramètres du modèle est répété  $n$  fois. Cependant, ce problème de recalcul peut être évité dans le cas d'un modèle de régression (linéaire en  $\mathbf{w}$ ) et une fonction de perte quadratique. En effet, on peut montrer que l'estimation précédente (5) peut s'écrire sous la forme plus simple

$$\begin{aligned}\widehat{\mathcal{E}} &:= \frac{1}{n} \sum_{i=1}^n (Y_i - f(\mathbf{X}_i, \widehat{\mathbf{w}}^{(-i)}))^2 \\ &= \frac{1}{n} \sum_{i=1}^n \left( \frac{Y_i - f(\mathbf{X}_i, \widehat{\mathbf{w}})}{1 - h_i} \right)^2,\end{aligned}\tag{6}$$

où  $\hat{\mathbf{w}} = (\mathbb{X}^\top \mathbb{X})^{-1} \mathbb{X}^\top \mathbf{Y}$  est l'estimateur des moindres-carrés utilisant toutes les données, et  $h_i$  est le  $i^{eme}$  élément diagonal de la matrice  $\mathbb{X}(\mathbb{X}^\top \mathbb{X})^{-1} \mathbb{X}^\top$ , pour tout  $i = 1, \dots, n$ .

- **Méthode 3 : ( $K$ -fold cross-validation ( $K$ -fold CV))**. Cette méthode est une alternative aux deux méthodes précédentes LOOCV et Apprentissage/Validation. Elle consiste à diviser de manière aléatoire le tableau des données en  $K$  groupes ("folds") de même effectif (ou presque). Notons  $G_1, \dots, G_K$  ces groupes. Le premier groupe est considéré comme un ensemble de validation, les  $K - 1$  autres groupes sont utilisés pour ajuster le modèle. Une première approximation de l'erreur théorique est donnée par

$$e_1 = \frac{1}{\text{Card}(G_1)} \sum_{i \in G_1} \ell(Y_i, f(\mathbf{X}_i, \hat{\mathbf{w}}^{(-1)})).$$

Cette procédure est répétée  $K$  fois, et à chaque fois, un groupe différent est traité comme ensemble de validation et les autres comme ensemble d'apprentissage ;

Ceci permet d'avoir  $K$  "estimations" de l'erreur théorique de prévision

$$e_k = \frac{1}{\text{Card}(G_k)} \sum_{i \in G_k} \ell(Y_i, f(\mathbf{X}_i, \hat{\mathbf{w}}^{(-k)})), \quad \forall k = 1, \dots, K.$$

L'estimation de l'erreur théorique de test par la K-fold CV est calculée en moyennant les valeurs précédentes

$$\begin{aligned} \widehat{\varepsilon} &:= \frac{1}{K} \sum_{k=1}^K e_k \\ &= \frac{1}{K} \sum_{k=1}^K \frac{1}{\text{Card}(G_k)} \sum_{i \in G_k} \ell(Y_i, f(\mathbf{X}_i, \hat{\mathbf{w}}^{(-k)})). \end{aligned} \quad (7)$$

## Remarque

Il n'est pas difficile de voir que la méthode  $K$ -fold CV = la méthode LOOCV si  $K = n$ . Dans la pratique, on applique une méthode  $K$ -fold CV souvent avec  $K = 10$ . L'avantage de ce choix est de réduire le temps de calcul pour l'estimation des paramètres du modèle, en particulier si  $p$  est grand.

## Statistique en grande dimension

- Lorsque le nombre de variables explicatives  $p$  est **grand**, les estimateurs MC du modèle RLM

$$Y = w_0 + w_1 X_1 + \cdots + w_p X_p + \varepsilon$$

possèdent généralement une **grande variance**.

### Idée des méthodes pénalisées

- Contraindre** la valeur des estimateurs MC de manière à **réduire la variance** (quitte à **augmenter un peu le biais**). **Comment ?**
- En imposant une **contrainte** sur les paramètres  $(w_1, \dots, w_p)^\top =: w$  du modèle :

$$\widehat{\mathbf{w}}_{\text{pen}} := \arg \min_{\mathbf{w}} \frac{1}{n} \|\mathbf{Y} - \mathbb{X}\mathbf{w}\|^2$$

sous la contrainte  $\|w\|? \leq b.$       (8)

## Questions

- Quelle norme utiliser pour la contrainte ?
- Existence/unicité des estimateurs  $\hat{\mathbf{w}}_{\text{pen}}$  ? Solutions explicites du problème d'optimisation ?
- Comment choisir le seuil  $b$  ?
  - si  $b$  est petit, les estimateurs  $\hat{\mathbf{w}}_{\text{pen}}$  sont trop contraints (donc proches de 0) ;
  - si  $b$  est grand, les estimateurs  $\hat{\mathbf{w}}_{\text{pen}}$  sont moins contraints (donc proches de estimateurs MC).

- La **régression ridge** consiste à minimiser le critère des moindres carrés pénalisé par la norme  $L_2$  du vecteur  $(w_1, \dots, w_p)$ .

## L'estimateur ridge

- L'estimateur ridge  $\hat{\mathbf{w}}^R$  est défini par

$$\hat{\mathbf{w}}^R := \arg \min_{\mathbf{w} \text{ t.q. } \sum_{j=1}^p w_j^2 \leq b} \frac{1}{2n} \sum_{i=1}^n \left( Y_i - w_0 - \sum_{j=1}^p w_j X_{i,j} \right)^2 \quad (9)$$

- ou de façon équivalente

$$\hat{\mathbf{w}}^R := \arg \min_{\mathbf{w} \in \mathbb{R}^{1+p}} \left\{ \frac{1}{2n} \sum_{i=1}^n \left( Y_i - w_0 - \sum_{j=1}^p w_j X_{i,j} \right)^2 + \lambda \sum_{j=1}^p w_j^2 \right\} \quad (10)$$

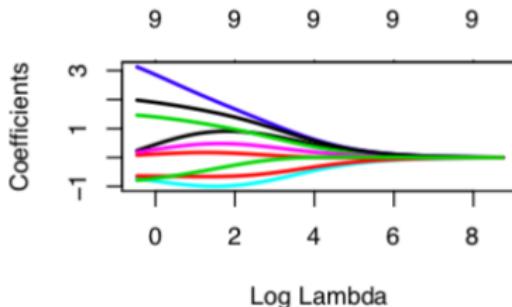
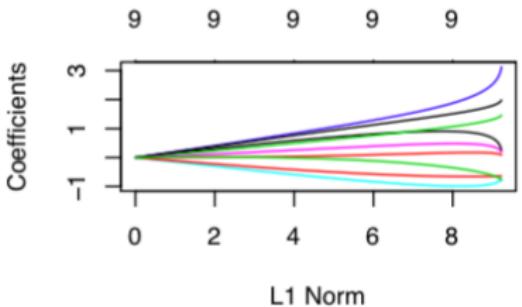
## Remarques

- Les définitions précédentes, (9) et (10), sont équivalentes dans le sens où pour tout  $b > 0$  il existe un unique  $\lambda > 0$  tels que les deux solutions coïncident ;
- Le coefficient  $w_0$  (de l'intercept) n'est pas pris en compte dans la pénalité ;
- L'estimateur ridge dépend évidemment du paramètre  $b$  (ou  $\lambda$ ) :  
 $\hat{\mathbf{w}}^R := \hat{\mathbf{w}}^R(b) := \hat{\mathbf{w}}^R(\lambda)$  ;
- Les variables explicatives sont le plus souvent réduites pour éviter les problèmes d'échelle dans la pénalité.

Sous R, le calcul des estimateurs ridge peut se faire à l'aide de la fonction 'glmnet()' du package 'glmnet' :

```
library(glmnet)

LAozoneData.mat <- as.matrix(LAozoneData)
reg.ridge <- glmnet(x = scale(LAozoneData.mat[,2:10]), y = LAozoneData.mat[,1], alpha = 0)
par(mfrow = c(2,2))
plot(reg.ridge, label = TRUE)
plot(reg.ridge, xvar = "lambda", label = TRUE, lwd = 2)
```



## Propriétés de l'estimateur ridge

- Lorsque les variables explicatives sont centrées-réduites, l'estimateur ridge s'écrit

$$\hat{\mathbf{w}}^R = \hat{\mathbf{w}}^R(\lambda) = (\mathbb{X}^\top \mathbb{X} + \lambda \mathbb{I})^{-1} \mathbb{X}^\top \mathbf{Y}.$$

- On en déduit

$$\text{biais}(\hat{\mathbf{w}}^R) = -\lambda (\mathbb{X}^\top \mathbb{X} + \lambda \mathbb{I})^{-1} \mathbf{w}$$

et

$$\text{Var}(\hat{\mathbf{w}}^R) = \sigma^2 (\mathbb{X}^\top \mathbb{X} + \lambda \mathbb{I})^{-1} \mathbb{X}^\top \mathbb{X} (\mathbb{X}^\top \mathbb{X} + \lambda \mathbb{I})^{-1}.$$

## Remarques

- Si  $\lambda = 0$ , on retrouve le biais (null) et la variance de l'estimateur MC ;
- Si  $\lambda$  augmente, le biais augmente et la variance diminue ;
- Si  $\lambda$  diminue, le biais diminue et la variance augmente ;
- Si  $\lambda \approx 0$ , alors  $\hat{\mathbf{w}}^R \approx \hat{\mathbf{w}}$  l'estimateur MC. Si  $\lambda$  est grand, alors  $\hat{\mathbf{w}}^R \approx \mathbf{0}$ .

## Choix de $\lambda$

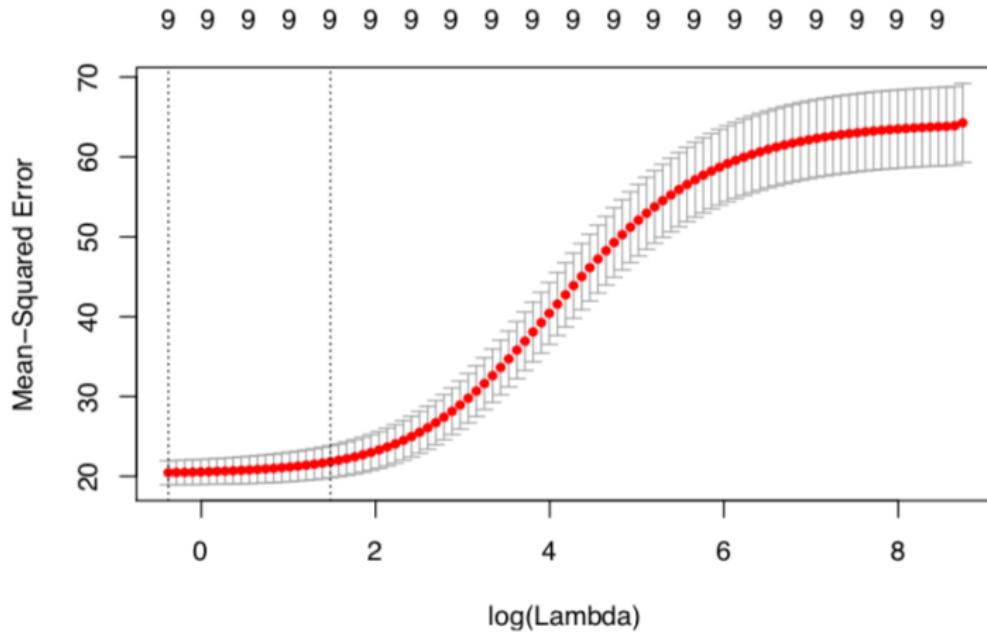
- Le choix de  $\lambda$  peut se faire, par validation croisée, de la manière suivante (sous R, on peut utiliser la fonction `cv.glmnet()`)
  - Estimation de l'erreur de prévision (par validation croisée) pour toutes les valeurs de  $\lambda$  ;
  - Choix de  $\lambda$  qui minimise l'erreur estimée.

```

reg.cvridge <- cv.glmnet(x = scale(LA ozoneData.mat[, 2:10]), y = LA ozoneData.mat[, 1],
                           alpha = 0)
bestlam <- reg.cvridge$lambda.min
bestlam

## [1] 0.6853818
plot(reg.cvridge)

```



```
min(reg.cvridge$cvm)

## [1] 20.46384

coef(reg.cvridge)

## 10 x 1 sparse Matrix of class "dgCMatrix"
##           1
## (Intercept) 11.7757576
## vh          0.8923481
## wind        0.1696268
## humidity   1.1056282
## temp        1.9604498
## ibh         -0.9910652
## dpg         0.4797878
## ibt         1.5518232
## vis         -0.6569836
## doy        -0.3909164
```

- La **régression lasso** consiste à minimiser le critère des moindres carrés pénalisé par la norme  $L_1$  du vecteur  $(w_1, \dots, w_p)$ .

## L'estimateur lasso (c.f. Tibshirani 1996)

- L'estimateur lasso  $\hat{\mathbf{w}}^L$  est défini par

$$\hat{\mathbf{w}}^L := \arg \min_{\mathbf{w} \text{ t.q. } \sum_{j=1}^p |w_j| \leq b} \frac{1}{2n} \sum_{i=1}^n \left( Y_i - w_0 - \sum_{j=1}^p X_{i,j} w_j \right)^2 \quad (11)$$

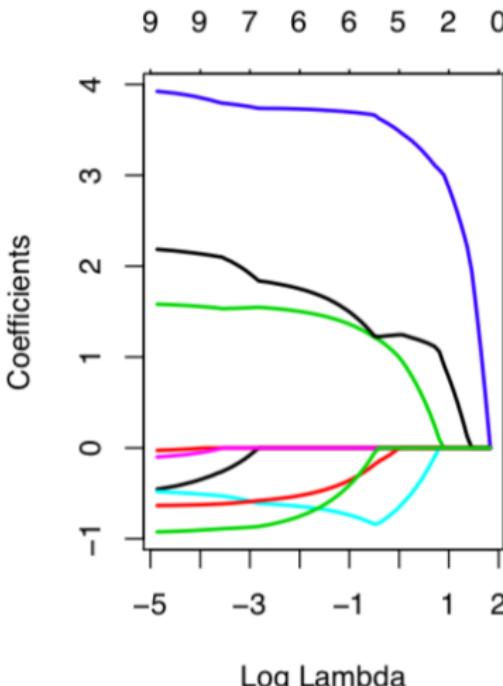
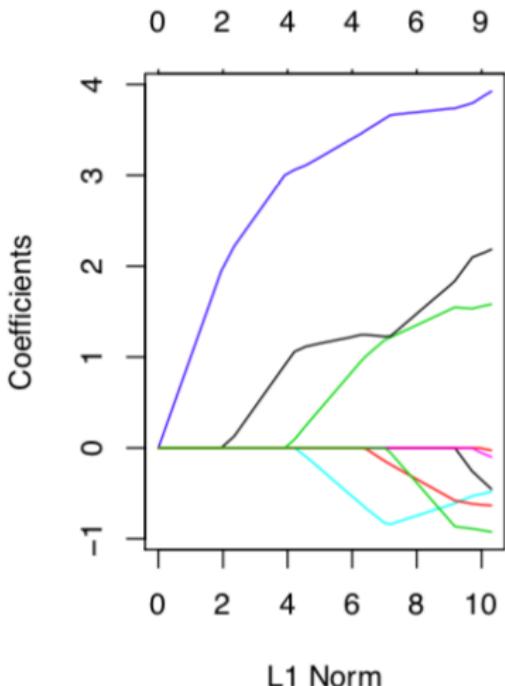
- ou de façon équivalente

$$\hat{\mathbf{w}}^L := \arg \min_{\mathbf{w} \in \mathbb{R}^{1+p}} \left\{ \frac{1}{2n} \sum_{i=1}^n \left( Y_i - w_0 - \sum_{j=1}^p X_{i,j} w_j \right)^2 + \lambda \sum_{j=1}^p |w_j| \right\} \quad (12)$$

## Remarques

- Comme pour la régression ridge, si  $\lambda$  augmente, alors le biais augmente et la variance diminue. Si  $\lambda$  diminue, le biais diminue et la variance augmente ;
- Comme pour la régression ridge, les variables explicatives sont le plus souvent réduites pour éviter les problèmes d'échelle dans la pénalité ;
- Le choix de  $\lambda$  peut se faire par validation croisée ;
- Contrairement au ridge, si  $\lambda$  augmente, alors le nombre de coefficients nuls augmente (c.f. Bühlmann and van de Geer, 2011). Donc **le lasso permet de faire de la sélection de variables** (contrairement au ridge).

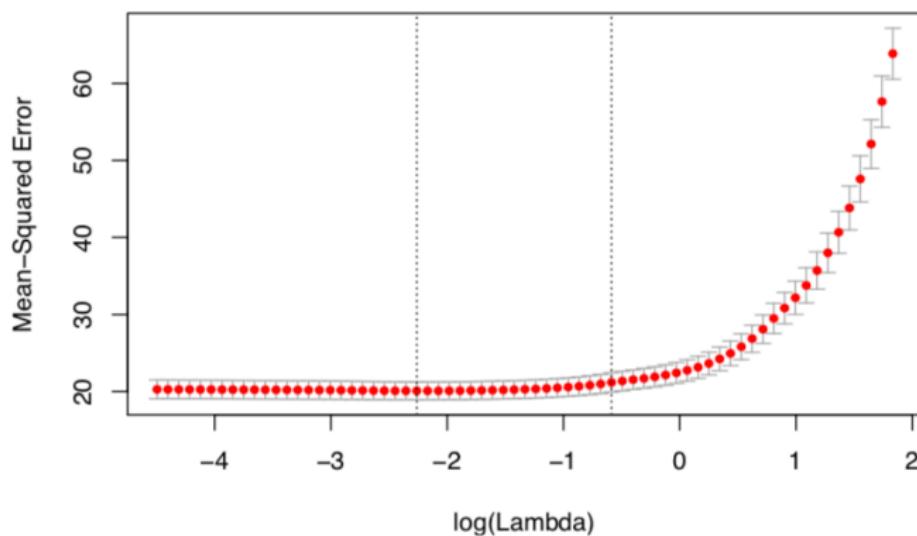
```
reg.lasso <- glmnet(x = scale(LA ozoneData.mat[, 2:10]), y = LA ozoneData.mat[, 1],
                     alpha = 1)
par(mfrow = c(1, 2))
plot(reg.lasso, label = TRUE)
plot(reg.lasso, xvar = "lambda", label = TRUE, lwd = 2)
```



## Choix de $\lambda$

```
reg.cvlasso <- cv.glmnet(x = scale(LA ozoneData.mat[,2:10]),  
                           y = LA ozoneData.mat[,1], alpha = 1)  
bestlam <- reg.cvlasso$lambda.min  
bestlam  
  
## [1] 0.1041719  
plot(reg.cvlasso)
```

9 9 8 7 7 6 6 6 6 6 5 5 4 4 2 1 0



```
min(reg.cvlasso$cvm)

## [1] 20.19825
coef(reg.cvlasso)

## 10 x 1 sparse Matrix of class "dgCMatrix"
##           1
## (Intercept) 11.7757576
## vh          .
## wind         .
## humidity    1.2484460
## temp         3.6709433
## ibh          -0.8178146
## dpg          .
## ibt          1.2876148
## vis          -0.2207656
## doy          -0.1320043
```

- Dans certains cas, les variables explicatives appartiennent à des groupes de variables prédéfinis ; c'est le cas par exemple des variables indicatrices des modalités d'une même variable (on veut les sélectionner toutes ou les exclure toutes).

## Définition : Group Lasso

En présence de  $d$  variables réparties en  $L$  groupes  $\mathbf{X}_1, \dots, \mathbf{X}_L$  de cardinal  $d_1, \dots, d_L$ . On note  $\mathbf{w}_\ell := (w_{\ell,1}, \dots, w_{\ell,d_\ell})$ , le vecteur des coefficients associé au groupe  $\mathbf{X}_\ell$ ,  $\ell = 1, \dots, L$ . Les **estimateurs group-lasso** s'obtiennent en minimisant le critère

$$\frac{1}{2n} \sum_{i=1}^n \left( Y_i - w_0 - \sum_{\ell=1}^L \mathbf{w}_\ell \mathbf{X}_{i,\ell} \right)^2 + \lambda \sum_{\ell=1}^L \sqrt{d_\ell} \|\mathbf{w}_\ell\|_2.$$

## Remarques

- Puisque

$$\|\mathbf{w}_\ell\|_2 = 0 \text{ ssi } w_{\ell,1} = \cdots = w_{\ell,d_\ell} = 0,$$

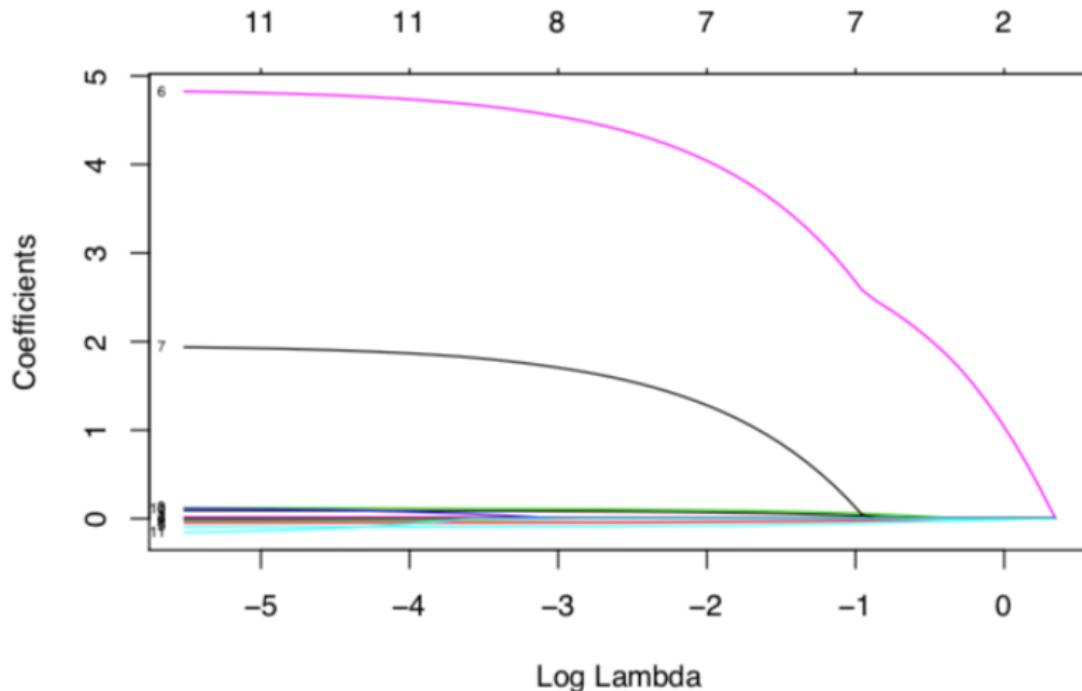
la méthode group-lasso encourage la mise à zéro des coefficients d'un même groupe ;

- La méthode peut se faire, sous R, à l'aide de la fonction 'gglasso()' du package 'gglasso'.

```
library(gglasso)
library(ISLR)
str(Carseats)

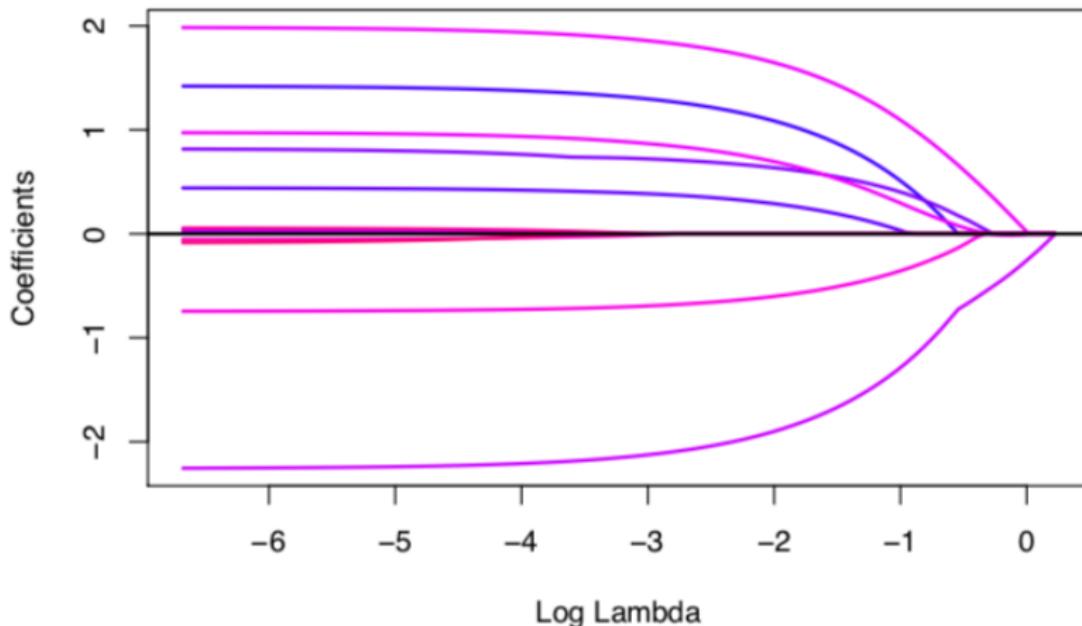
## 'data.frame':   400 obs. of  11 variables:
## $ Sales      : num  9.5 11.22 10.06 7.4 4.15 ...
## $ CompPrice  : num  138 111 113 117 141 124 115 136 132 132 ...
## $ Income     : num  73 48 35 100 64 113 105 81 110 113 ...
## $ Advertising: num  11 16 10 4 3 13 0 15 0 0 ...
## $ Population : num  276 260 269 466 340 501 45 425 108 131 ...
## $ Price      : num  120 83 80 97 128 72 108 120 124 124 ...
## $ ShelveLoc  : Factor w/ 3 levels "Bad","Good","Medium": 1 2 3 3 1 1 3 2 3 3 ...
## $ Age        : num  42 65 59 55 38 78 71 67 76 76 ...
## $ Education  : num  17 10 12 14 13 16 15 10 10 17 ...
## $ Urban      : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 1 1 ...
## $ US         : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 1 2 1 2 ...
```

```
D <- model.matrix(Sales ~ ., data = Carseats)[,-1]
model <- glmnet(x = D, y = Carseats$Sales, alpha = 1)
plot(model,label = TRUE, xvar = "lambda")
```



#on définit les groupes de variables

```
groupe <- c(1,2,3,4,5,6,6,7,8,9,10)
model1 <- gglasso(x = scale(D), y = Carseats$Sales, group = groupe)
plot(model1)
```



```
library(boot)
reg.cv.grouplasso <- cv.gglasso(x = scale(D), y = Carseats$Sales, group = groupe)
reg.cv.grouplasso$lambda.min

## [1] 0.003827857
min(reg.cv.grouplasso$cvm)

## [1] 1.053357
coef(reg.cv.grouplasso)

##                1
## (Intercept) 7.4963250
## CompPrice   1.2297180
## Income      0.3550674
## Advertising 0.6937029
## Population  0.0000000
## Price       -2.0540540
## ShelveLocGood 1.7899039
## ShelveLocMedium 0.8113856
## Age         -0.6644544
## Education   0.0000000
## UrbanYes    0.0000000
## USYes       0.0000000
```

```
model.RLM <- glm(Sales ~ . -Population-Education-Urban-US, data = Carseats)
cv.glm(data = Carseats, glmfit = model.RLM, K = 10)$delta[2]
```

```
## [1] 1.065201
```

```
model.RLM <- glm(Sales ~ ., data = Carseats)
cv.glm(data = Carseats, glmfit = model.RLM, K = 10)$delta[2]
```

```
## [1] 1.080218
```

## Références :

- Bruce, Peter, and Andrew Bruce. 2017. Practical Statistics for Data Scientists;
- Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58 : 267-288 ;
- Cornillon, P. A. et al. (2018). R pour la statistique et la science des données. Presses universitaires de Rennes.  
“<https://r-stat-sc-donnees.github.io>” ;
- Friedman, J., Hastie, T. & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). New York, NY, USA : Springer series in statistics.  
“<https://web.stanford.edu/~hastie/ElemStatLearn/>” ;

- James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112). New York : Springer. “<http://www-bcf.usc.edu/gareth/ISL/>” ;
- Saporta, G. (2006). Probabilités, analyse des données et Statistique. Editions TECHNIP ;
- Sites web : “<http://www.rdatamining.com/home>”, “<https://perso.univ-rennes2.fr/laurent.rouviere>” ; “<http://eric.univ-lyon2.fr/ricco/cours/index.html>”, “<http://www.sthda.com/french/>”, “<http://www.statsoft.com/Textbook>”, ...