

Livret : Alternatives aux MCO

Sabaye Fried-Junior

3 février 2021



Table des matières

- ① Famille des méthodes standards
 - ▶ Moindres carrés ordinaires
 - ▶ Régression de Deming
 - ▶ Régression orthogonale
- ② Famille des méthodes robustes
 - ▶ Régression quantile
 - ▶ Estimateur de Theil-Sen
- ③ Famille des méthodes de Machine Learning
 - ▶ Régression Ridge
 - ▶ Régression Lasso
- ④ Famille des méthodes non-paramétriques
 - ▶ Régression LOESS
 - ▶ Régression Kernel

Famille des méthodes standards

- Moindres carrés ordinaires
- Régression de Deming
- Régression orthogonale

Moindres carrés ordinaires

- **Legendre** : *"Nouvelle méthodes pour la détermination des orbites et des comètes"*, 1805
- **Gauss** : *"THEORIA MOTVS CORPORVM COELESTV"*, 1809

- Considérons tout d'abord un modèle de régression linéaire simple :

$$y = \beta_0 + \beta_1 X + e$$

où :

- y est la variable à expliquer et X est la variable explicative du modèle. Les paramètres du modèle ou encore coefficients de régression, β_0 et β_1 sont respectivement l'ordonnée à l'origine et la pente de la droite de régression associée à cette équation. Enfin, e représente la différence entre les vraies valeurs et les valeurs observées de la y . Pour des raisons d'inférences statistiques, on affirme que e est une variable aléatoire I.I.D tel que $e \sim N(0, \sigma^2)$

- y est considérée comme une variable aléatoire avec :
 $\mathbb{E}(y) = \beta_0 + \beta_1 X$ et $\text{var}(y) = \sigma^2$
- Parfois X peut être considéré également comme une variable aléatoire.
 Dans ce cas, on prend en compte la moyenne et la variance conditionnelle de y sachant $X = x$:
 $\mathbb{E}(y | x) = \beta_0 + \beta_1 x$ et $\text{var}(y | x) = \sigma^2$
- Les paramètres β_0 , β_1 et σ^2 sont inconnus et e est inobservables.
- Afin de déterminer une estimation de ces paramètres, l'une des techniques couramment utilisée est celle des **moindres carrés ordinaires**.

- On observe un échantillon de N ensembles d'observations (x_i, y_i) ($i = 1, \dots, N$). Nous pouvons écrire l'équation précédente comme suit :

$$y_i = \beta_0 + \beta_1 x_i + e_i$$

- Le principe de la méthode des moindres carrés ordinaires consiste à estimer les paramètres β_0 et β_1 de sorte que la somme des carrés de la différence verticale entre les observations et la droite de régression soit minimale.

- On minimise la quantité suivante afin d'avoir une estimation des coefficients :

$$S(\beta_0, \beta_1) = \sum_{i=1}^N (e_i)^2 = \sum_{i=1}^N (y_i - \beta_0 - \beta_1 x_i)^2$$

- En égalisant l'équation précédente à 0 on obtient donc les solutions de β_0 et β_1 qui sont les estimateurs des moindres carrés ordinaires de β_0 et β_1 . Ces estimateurs sont :
- $b_0 = \bar{y} - b_1 \bar{x}$
- $b_1 = \frac{\text{cov}[X, Y]}{\sigma_x^2}$

Régression linéaire multiple

- Un modèle de régression linéaire multiple est une extension du modèle de régression linéaire simple dans le sens où il y a plusieurs variables explicatives.
- L'équation de régression pour la i -ème observation s'écrit :

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \epsilon_i = \beta_0 + \sum_{j=1}^p \beta_j x_{ij} + \epsilon_i$$

où :

- $i = 1, \dots, N$ correspond au nombre d'observations
- y_i est la i -ème observation de la variable dépendante y
- $x_{i,j}$ est la i -ème observation de la k -ème variable avec $k = 0, \dots, J$
- ϵ_i : erreur du modèle (v.a.r) (part de la variabilité de Y qui n'est pas expliquée par le lien fonctionnel linéaire)

Régression linéaire multiple

- Pour tous les individus N , le modèle peut s'écrire sous la forme matricielle :

$$\mathbf{y} = \mathbf{X} \boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

- \mathbf{y} est un vecteur de dimensions $[N \times 1]$
- \mathbf{X} est une matrice de dimensions $[N \times (J + 1)]$
- $\boldsymbol{\beta}$ est un vecteur de dimensions $[(J+1) \times 1]$
- $\boldsymbol{\varepsilon}$ est un vecteur de dimensions $[N \times 1]$

Régression linéaire multiple

Estimateurs

La méthode standard d'estimation des paramètres utilisée ici est la même que dans le cas de la régression linéaire simple à savoir : les **moindres carrés ordinaires** :

$$S(\beta_0, \dots, \beta_p) = \sum_{i=1}^n (\epsilon_i)^2 = \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij})^2 = \|\epsilon\|^2$$

Après résolution, on a :

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

Propriétés des estimateurs

- $E[\hat{\beta}] = \beta$ estimateur sans biais
- $V[\hat{\beta}] = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}$

Régression de Deming

- **R.J.Carroll et David Ruppert** : *"The use and misuse of orthognal regression in linear errors in variables models"*, The American Statistician, Février 1996, Vol.50, No.1

Régression de DEMING

Les modèles d'erreur de mesure supposent que la variable réponse et qu'une ou plusieurs des variables prédictives font l'objet d'erreur de mesure.

De manière générale, nous avons :

$$Y = y_{true} + \varepsilon \quad (1)$$

$$W = X + U \quad (2)$$

- Y = Variable à expliquer observée, $Y = y_{true} + \varepsilon$, $\text{var}(\varepsilon) = \sigma_{\varepsilon}^2$
- y_{true} = "Vraie" valeur de la variable à expliquer
- W = Prédicteur observé, $W = X + U$, $\text{var}(U) = \sigma_u^2$
- X = "Vraie" valeur du prédicteur
- ε et U sont indépendants

Régression de DEMING

- En combinant (1) et (2) nous avons le modèle suivant :

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

- Un modèle de régression de DEMING requiert la connaissance du ratio des variances :

$$\eta = \frac{\text{var}(Y|X)}{\text{var}(W|X)} = \frac{\sigma_v^2}{\sigma_u^2}$$

Régression de DEMING

- L'estimateur de la régression de DEMING s'obtient en minimisant la quantité :

$$\sum_{i=1}^N \{ (Y_i - \beta_0 - \beta_1 X_i)^2 / \eta + (W_i - X_i)^2 \} \quad (3)$$

- $\beta_0, \beta_1, X_1, \dots, X_n$ sont des inconnus.
 - Si $\eta = 1$, alors (3) sera égale à la distance orthogonale de $(Y_i, W_i)_{i=1}^n$ de la droite $(\beta_0 + \beta_1 X_i, X_i)_{i=1}^n$.
 - Si $\eta \neq 1$ alors (3) est une distance orthogonale pondérée.
- L'estimateur de la régression de DEMING est donc :

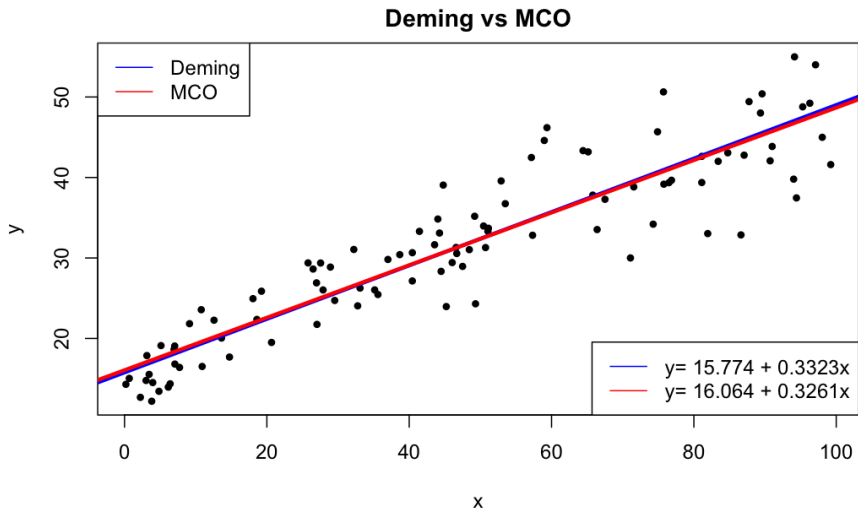
$$\beta_1(\text{OR}) = \frac{s_y^2 - \eta s_w^2 + \{ (s_y^2 - \eta s_w^2)^2 + 4\eta s_{wy}^2 \}^{1/2}}{2s_{wy}}$$

Avec : $\text{var}(Y) = s_y^2$; $\text{var}(W) = s_w^2$; $\text{Cov}(WY) = s_{wy}$

La fonction `mcr()`

- Package nécessaire : `mcr`
- Définition du modèle :
`Modèle <- mcreg(Y, X, method.reg = "Deming")`
- Résultats des estimations :
`Modèle@para`
- Représentation graphique :
`abline(Modèle@para[1 :2], col = "...", ...)`

MCO Vs Deming



Régression orthogonale

Le principe des moindres carrés dans la régression orthogonale est la minimisation de la distance perpendiculaire au carré entre les points et la droite de régression afin d'obtenir une estimation des coefficients de régression.

Nous avons le modèle suivant :

$$Y_i = \beta_0 + \beta_1 X_i$$

qui définit

$$E_i = Y_i - \beta_0 - \beta_1 X_i = 0 \quad (1)$$

Régression orthogonale

Les coefficients de régression sont obtenus en minimisant la distance perpendiculaire au carré entre les points (x_i, y_i) ($i = 1, \dots, N$) :

$$d_i^2 = (X_i - x_i)^2 + (Y_i - y_i)^2 \quad (2)$$

L'objectif étant de minimiser $\sum_{i=1}^N d_i^2$ pour obtenir les estimations de β_0 et de β_1 .

Les coefficients de régression sont obtenus en minimisant (2) sous la contrainte (1) en utilisant la méthode du multiplicateur de Lagrange.

La fonction de Lagrange étant :

$$L = \sum_{i=1}^N d_i^2 - 2 \sum_{i=1}^N \lambda_i E_i$$

où $\lambda_1, \dots, \lambda_i$ sont les multiplicateurs de Lagrange.

Régression orthogonale

Après résolution nous obtenons une estimation de régression orthogonale de β_0 et β_1 tel que :

$$\hat{\beta}_{0OR} = \bar{y} - \hat{\beta}_{1OR}\bar{x}$$

et

$$\hat{\beta}_{1OR} = \frac{(SYY - SXX) + \text{sgn}(SXY)\sqrt{(SXX - SYY)^2 + 4SXY}}{2SXY}$$

Où :

- $\text{sgn}(SXY)$ est le signe de (SXY)
- SXX et SYY respectivement $\text{Var}(X)$ et $\text{Var}(Y)$
- $SXY = \text{cov}(XY)$

Famille des régressions robustes

- Régression quantile
- Estimateur de Theil-Sen

Régression quantile

- **Roger Koenker et Gilbert Bassett** : "*Regression quantile*", *Econometrica*, Vol.46, No.1, Janvier 1978

Régression Quantile

Un des problèmes avec la régression linéaire classique est la sensibilité des coefficients aux valeurs extrêmes. Ce fait est dû à la dépendance des estimateurs à la **moyenne empirique**.

$$\hat{\alpha} = \bar{Y} - \beta \bar{X}$$
$$\hat{\beta} = \frac{\sum_{i=1}^N (x_i - \bar{X})(y_i - \bar{Y})}{\sum_{i=1}^N (x_i - \bar{X})^2}$$

Une manière alternative d'estimer un modèle linéaire est d'utiliser des estimateurs β qui ne dépendent pas de la moyenne empirique comme les estimateurs MCO. La régression quantile permet de faire des régressions sur des quantiles précis des variables explicatives.

Pour un modèle $Y = X\beta + \epsilon$,

La fonction objectif à minimiser pour une regression au θ^{me} quantile est :

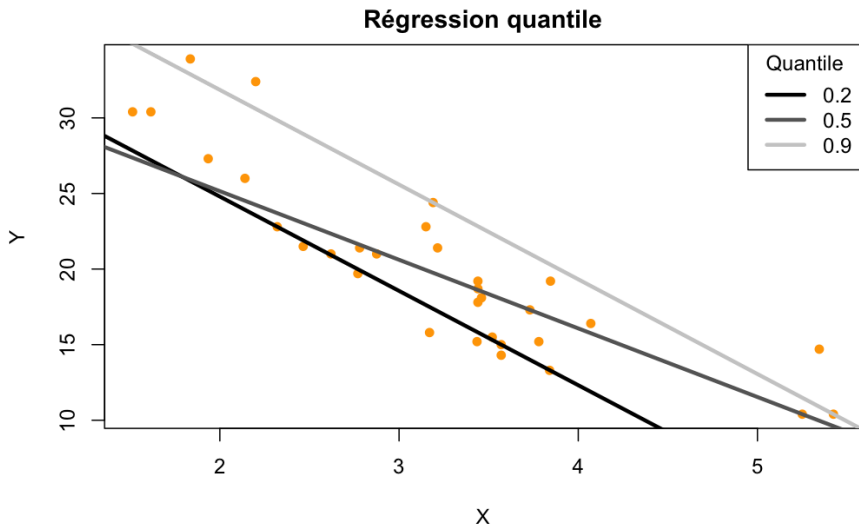
$$\text{Min}\{\sum_{Y_i \leq \beta X_i} \theta |Y_i - \beta X_i| + \sum_{Y_i > \beta X_i} (1 - \theta) |Y_i - \beta X_i|\}$$

- On ne minimise pas le carré des résidus mais la valeur absolue de l'erreur, pondérée par la pénalité θ
- On peut décider de régresser un échantillon sur plusieurs quantiles pour mieux rendre compte de la dispersion sur chaque partie de la population.
- Dans des cas où la variance du terme d'erreur n'est pas constante (hétéroscedasticité), cette méthode fournit de meilleures estimations que les MCO.

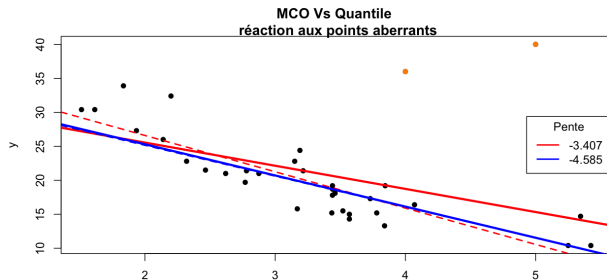
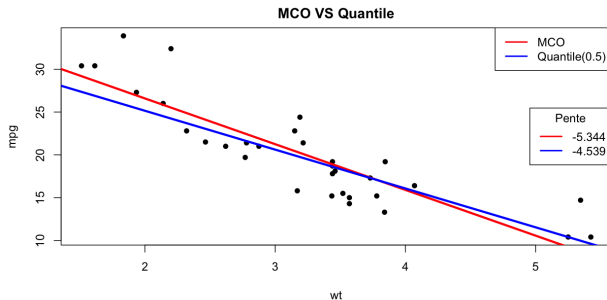
La fonction rq()

- Package nécessaire : quantreg
- Définition du modèle :
`Modèle <- rq(Y ~ X1, ..., Xn, dataframe, tau = c(0.1, 0.5, 0.9))`
- Représentation graphique :
`plot(Y ~ X, dataframe, ...)`
`abline(rq(Y ~ X, dataframe, tau = "..."), ...)`

Exemple



MCO Vs Régression quantile



Estimateur de Theil - Sen

- **Xin Dang, Hanxiang Peng, Xueqin Wang and Heping Zhang :**
"Theil-Sen Estimators in a Multiple Linear Regression Model",
University of Mississippi and Yale University, 2009

Estimateur de Theil - Sen

L'estimateur Theil-Sen d'un ensemble de points bidimensionnels (x_i, y_i) est la médiane des pentes $\frac{(y_j - y_i)}{(x_j - x_i)}$ déterminé par toutes les paires de points de l'échantillon.

Cet estimateur est très peu sensible aux valeurs aberrantes et donc est nettement plus précis que les MCO dans le cadre la régression linéaire simple pour les données asymétriques et hétéroscédastiques.

Considérons un modèle de régression linéaire simple :

$$y_i = \beta_0 + \beta_1 x_i + e_i$$

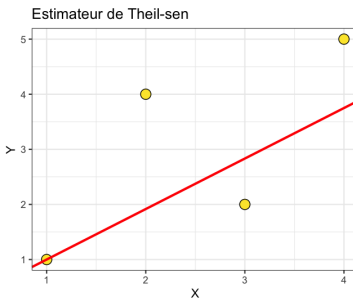
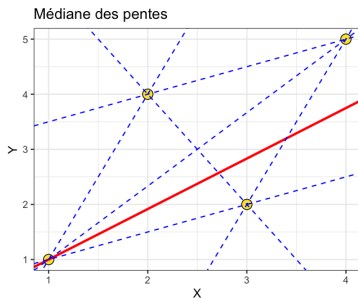
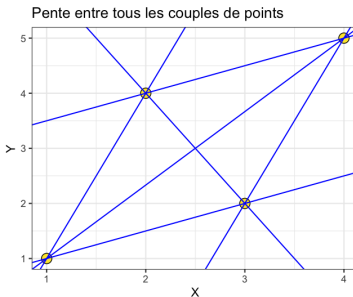
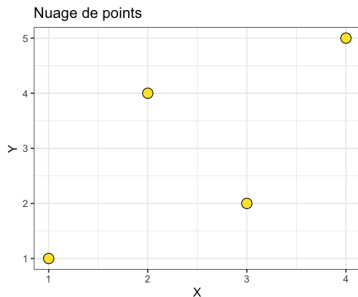
Un estimateur robuste de la pente est la médiane de ces estimations par moindres carrés :

$$\tilde{\beta}_1 = \text{Med} \left\{ \hat{\beta}_{1i,j} = \frac{y_i - y_j}{x_i - x_j} : x_i \neq x_j, 1 \leq i \leq j \leq n \right\}$$

De même, β_0 peut être estimée, par la médiane des estimations par moindres carrés :

$$\tilde{\beta}_0 = \text{Med} \left\{ \hat{\beta}_{0i,j} = \frac{y_j x_i - y_i x_j}{x_i - x_j} : x_i \neq x_j, 1 \leq i \leq j \leq n \right\}$$

Construction graphique

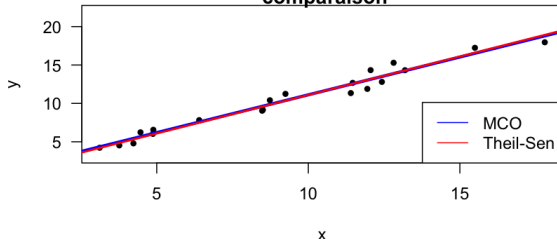


La fonction `mblm()`

- Package nécessaire : `mblm`
- Définition du modèle :
`Modèle <- mblm(Y ~ X1, ..., Xn, dataframe, repeated = TRUE)`
- Représentation graphique :
`ggplot(dataframe, aes(X, Y)) + geom_abline(intercept = coef(ts_fit)[1], slope = coef(ts_fit)[2], col = "..", ...)`

MCO Vs Theil-sen

MCO Vs Theil-Sen :
comparaison



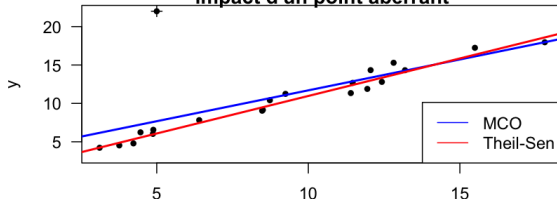
MCO

$$R^2 = 0.9595$$

Theil-Sen

$$\text{pente} = 0.91667$$

MCO Vs Theil-Sen :
impact d'un point aberrant



MCO

$$R^2 = 0.389$$

Theil-Sen

$$\text{pente} = 0.9774$$

Famille des méthodes de machine Learning

- Régression Ridge
- Régression Lasso

Régressions Ridge

- **Robert Tibshirani** : *"Regression shrinkage and selection via the Lasso"*, Journal of the Royal statistical society, 1996.
- **Arthur E.Hoerl, Robert W.Kennard** : *"Ridge regression : biased estimation for none orthogonal problems"*, Technometrics, Février 1970.

- Imaginons que nous voulions créer un modèle prédictif que nous entraînons sur un échantillon de données d'entraînement, pour prédire un jeu de données test : problématique de **Machine Learning**.
- Dans certains cas (peu de données d'entraînement, ...), entraîner un modèle linéaire par la minimisation du carré des résidus mène au surapprentissage, résultant en de piètres prédictions sur les données test.
- La méthode de la régression Ridge instaure une pénalité de biais lors de la phase d'apprentissage pour réduire la variance du modèle final.

La régression Ridge consiste à minimiser la somme des carrés des résidus, plus une pénalité sur les coefficients à estimer. Formellement :

$$\text{Min}\{\sum_{i=1}^N e_i^2 + \lambda \cdot \beta^2\}$$

- Cette fonction objectif à minimiser dépend positivement des paramètres à estimer β , permet donc de limiter l'ordre de grandeur des $\hat{\beta} \rightarrow$ Pénalité sur le biais des coefficients, contrôlé par le paramètre λ
- En pratique, on détermine de manière ad-hoc : On choisit par **validation croisée** la valeur de λ qui minimise la variance du modèle final.
- Permet bien souvent d'obtenir une réduction considérable de la variance du modèle, par rapport aux MCO, lorsque confronté aux données test.

Régressions Lasso

- La régression Lasso est très proche de la méthode Ridge.
- La différence superficielle réside dans la modification de la fonction objectif :

$$\text{Min}\{\sum_{i=1}^N e_i^2 + \lambda \cdot |\beta|\}$$

→ Pénalité différente : on utilise la valeur absolue des coefficients.
→ Mathématiquement, permet de se séparer de variables inutiles (i.e $\hat{\beta}_i = 0$) pour minimiser cette fonction. Cette méthode a l'avantage supplémentaire d'opérer un tri sur les variables.

Comparaison Ridge & Lasso

Ridge ou Lasso ?

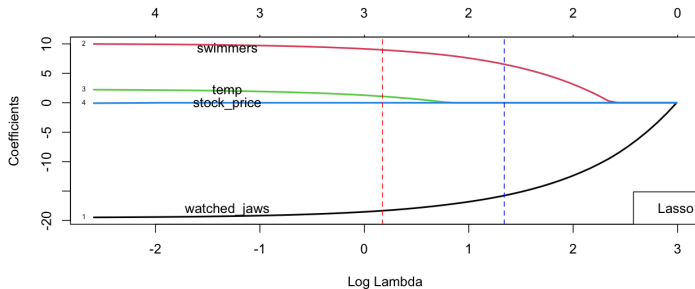
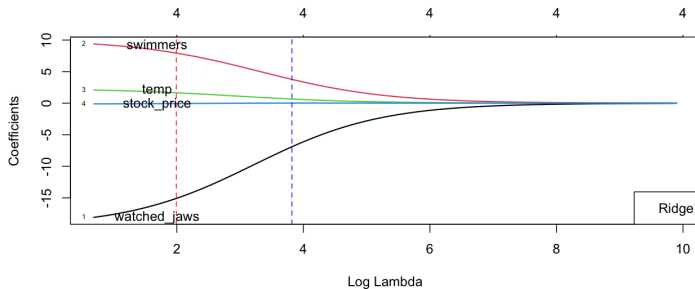
→ Ces deux méthodes ne sont pas équivalentes :

- La régression Ridge fait de meilleures prédictions lorsque le modèle est correctement spécifié au départ ($\hat{\beta}_i \neq 0, \forall i$)
- La régression Lasso est plus flexible, et fait de meilleures prédictions a priori, lorsque l'on ne sait pas si les régresseurs expliquent significativement la variable d'intérêt ou non.
- Ces deux méthodes sont donc complémentaires : On peut opérer un tri via Lasso pour en sortir une équation ne comportant que des variables significatives, et entrainer un nouveau modèle par la méthode Ridge.

Application sur Rstudio

- Package nécessaire : glmnet
- Préparation des données :
`Matrice <- model.matrix(Y ~ -1, dataframe)`
`Response <- dataframe$Y`
- Définition des modèles :
`Lasso <- glmnet(scale(Matrice), response, alpha=1)`
`Ridge <- glmnet(scale(Matrice), response, alpha=0)`
- Résultats des estimations :
`Modèle@para`
- Représentation graphique :
`plot(modèle, xvar = "lambda", label=T)`
`abline(modèle@para[1 :2], col = "...", ...)`

Ridge Vs Lasso



Famille des régressions non-paramétriques

- Régression LOESS
- Estimateur Kernel

Régression non paramétrique

- **Clive Loader** : "*Local regression and Likelihood*", 1999.

Cadre général

Dans un modèle de régression non-paramétrique, la fonction de lien entre \mathbf{X} et \mathbf{Y} n'a pas de forme explicite et ne peut pas s'écrire en fonction d'un nombre réduit de paramètres. On cherche :

$$y = f(x) + \epsilon, \text{ où } E(Y|X = x) = f(x)$$

Avec une approche non paramétrique on aboutit à :

- une relation graphique entre \mathbf{Y} et \mathbf{X}
- des estimateurs (smoothers) beaucoup plus souples grâce au peu d'hypothèses que nous gardons
- Il n'existe pas de forme analytique de la fonction de lien $f(x)$

Régression kernel

- La régression avec lissage par opérateur à noyau ou régression kernel cherche à estimer la fonction $f(x_i)$ en tout point x_1, x_2, \dots, x_N . Pour cela on utilise communément le lissage par opérateur à noyau ou kernel smoother (Nadaraya, 1964 et Watson, 1964).
- L'estimateur à noyau (kernel estimate) de la fonction de lien évaluée au point x_0 , noté $f(x_0)$, est défini par :

$$f(x_0) = \sum_{i=1}^N w_i(x_0) y_i$$

avec :

$$w_i(x_0) = \frac{K\left(\frac{x_i - x_0}{\lambda}\right)}{\sum_{i=1}^N K\left(\frac{x_i - x_0}{\lambda}\right)}$$

Où $K(\cdot)$ désigne une fonction kernel, $\lambda > 0$ un paramètre de lissage (bandwidth parameter) et N la taille de l'échantillon utilisée pour l'estimation

- **Remarque 1** : La fonction de lien évaluée au point de x_0 est donc définie comme une somme pondérée des observations y_i dont les poids $w_i(x_0)$ dépendent de x_0 .
- **Remarque 2** : La fonction $w_i(x_0)$ définit le poids qui doit être attribué au couple d'observations (x_i, y_i) dans la valeur de la fonction de lien évaluée au point d'abscisse x_0 . Généralement, plus les points x_i sont proches de x_0 , plus le poids sera important : $w_i(x_0)$ est donc décroissante dans la distance $|x_0 - x_i|$

Ces poids dépendent de fonction kernel (ou opérateur à noyau) qui correspond tout simplement à des fonctions de densité de probabilité.

Une fonction de kernel $K(\frac{x_i - x_0}{\lambda}) = K(u)$ vérifient les propriétés suivantes :

- (i) $K(u) \geq 0$
- (ii) $K(u)$ est normalisé de sorte que : $\int K(u) du = 1$
- (iii) $K(u)$ atteint son maximum en 0 lorsque $x_i = x_0$ et décroît avec la distance $|x_0 - x_i|$.
- (iv) $K(u)$ est symétrique : le kernel ne dépende que de la distance $|x_0 - x_i|$ et non du signe de $x_0 - x_i$.

Régressions locales : LOESS

Un des problèmes essentiels avec la régression Kernel réside dans le manque de robustesse de ces estimateurs pour les valeurs extrêmes de \mathbf{X} . Une solution alternative, plus robuste pour les valeurs extrêmes consiste en l'utilisation de régression locales.

Le principe général d'une **régression locale** est de postuler que la fonction de lien $f(x_0)$ évaluée au point x_0 peut être approximé par la valeur d'une fonction paramétrique évalué localement au voisinage $N(x_0)$ du point de référence x_0 .

La **procédure LOESS** attribue des poids selon une fonction de type tri cubique en fonction de la distance au centre de classe.

La fonction de pondération généralement utilisée pour effectuer une régression locale est une fonction cubique pondérée :

$$w(x) = (1 - |x|^3)^3 / [|x| < 1]$$

Application sur Rstudio

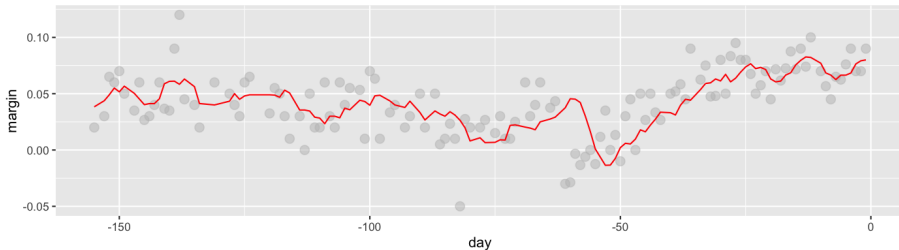
- Package nécessaire : tidyverse, dslabs, ggplot2
- Définition des modèles :

```
Kernel <- with(dataframe, ksmooth(Y, margin, kernel = "box",  
bandwidth = span))  
LOESS <- loess(Y ~ X, degree=1, span = span, dataframe)
```
- Représentation graphique, avec ggplot :

```
dataframe %>% mutate(smooth = Kernel$Y) %>% ggplot(aes(Y,  
X)) + geom_point() + geom_line(aes(Y, smooth))
```

Illustration

Kernel



LOESS

