Réseaux de neurones : modèle linéaire

Fabrice Rossi

http://apiacoa.org/contact.html.

Université Paris-IX Dauphine

Plan du cours “modèle linéaire”

1. Le modèle linéaire (lien avec les réseaux de neurones)
2. La régression linéaire simple
3. La régression linéaire multiple
4. Comportement asymptotique des moindres carrés
5. Modèle linéaire et discrimination

Cadre applicatif

Le modèle linéaire s'applique aux problèmes **supervisés** :

- Régression
- Discrimination (Classification supervisée)

Rappel du modèle mathématique :

- N exemples, des vecteurs $(x^k)_{1 \leq k \leq N}$ de $\mathbb{R}^n$
- discrimination : chaque vecteur est associé à une classe
- régression : chaque vecteur est associé à une cible, les $(y^k)_{1 \leq k \leq N}$, des vecteurs de $\mathbb{R}^p$

Expression statistique des problèmes :

- Discrimination : estimer $P(C_j|x)$
- Régression : estimer $E(Y|x)$

Bibliographie : voir les excellents supports de cours de Philippe Besse, à l'url :

`http://www.lsp.ups-tlse.fr/Besse/enseignement.html`

Le modèle linéaire en régression

On cherche donc à trouver un lien entre x^k et y^k, c'est-à-dire trouver une fonction f telle que $f(x^k) \simeq y^k$. Le modèle le plus simple est le **modèle linéaire** :

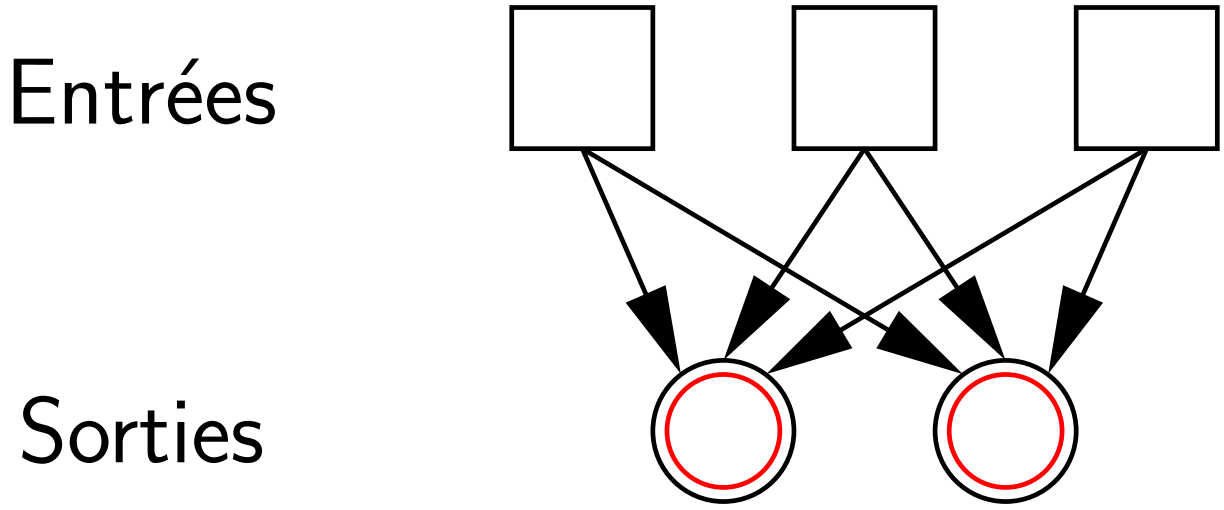
- paramètres : une matrice A (p lignes et n colonnes) et un vecteur b (dans $\mathbb{R}^p$)
- le modèle : $f(x^k) = Ax^k + b$
- le modèle est :
  - affine par rapport à x^k
  - linéaire par rapport à (A, b)
- Comment choisir A et b ?

Lien avec les réseaux de neurones

Un réseau à une couche :

- structure la plus simple possible (ce n'est même pas un réseau !)
- très proche des méthodes classiques (linéaires)

Graphiquement :



Ici, 3 entrées et 2 sorties : une fonction de $\mathbb{R}^3$ dans $\mathbb{R}^2$.

Neurone élémentaire

Un neurone est en général défini par :

1. une fonction d'activation (ou de transfert), $T : \mathbb{R} \to \mathbb{R}$
2. un nombre entier d'entrées

Un neurone à n entrées est associé à $n + 1$ paramètres numériques :

- n poids synaptiques (chaque poids correspond à une flèche dans le dessin)
- un seuil (*threshold*) ou biais (*bias*)

Un neurone à n entrées calcule la fonction suivante :

$$f(x_1, \ldots, x_n) = T\left(\sum_{i=1}^{n} a_i x_i + b\right)$$

Calcul pour une couche

Un réseau à une couche avec n et p sorties calcule donc :

$$f(x_1, \ldots, x_n)_j = T_j \left(\sum_{i=1}^{n} a_{ji} x_i + b_j \right)$$

soit matriciellement :

$$f(x) = \mathbb{T}(Ax + b),$$

avec

$$\mathbb{T}(y) = \begin{pmatrix} T_1(y_1) \\ \cdots \\ T_p(y_p) \end{pmatrix}$$

Si $T = Id$, on obtient le modèle linéaire !

Régression linéaire simple

Le cas le plus simple de régression linéaire correspond à $n = p = 1$:

- on dispose de N couples de réels $(x^k, y^k)_{1 \leq k \leq N}$
- on cherche a et b deux réels tels que $y^k \simeq ax^k + b$
- solution classique, les **moindres carrés** :
  - on cherche a et b qui minimisent

$$\widehat{\mathcal{E}}(a, b) = \sum_{k=1}^{N} \left(ax^k + b - y^k\right)^2$$

  - on a

$$\frac{\partial \widehat{\mathcal{E}}}{\partial b} = 2N \left(b + \frac{1}{N} \sum_{k=1}^{N} (ax^k - y^k) \right)$$

Régression linéaire simple (2)

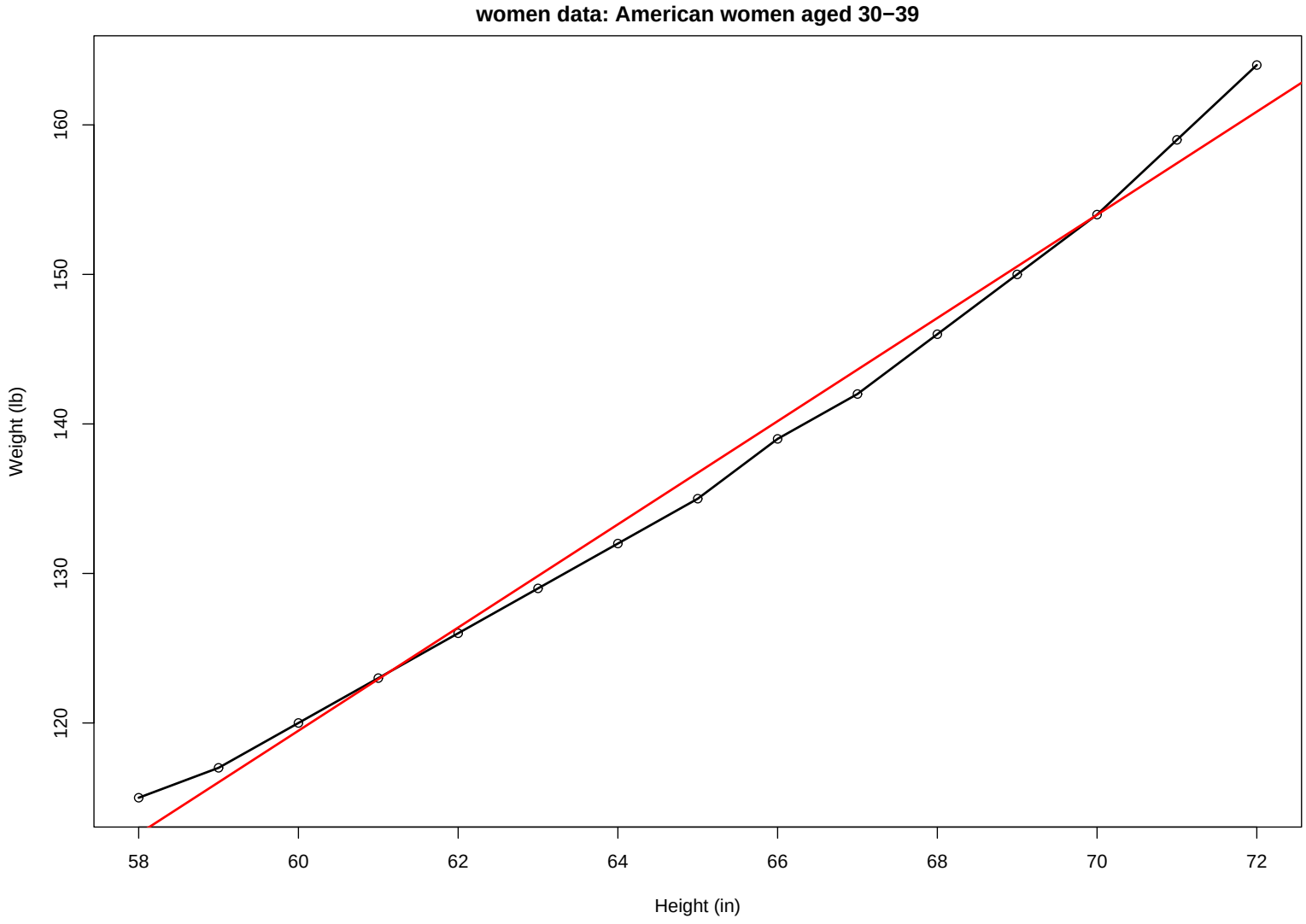
- on a

$$\frac{\partial \widehat{\mathcal{E}}}{\partial a} = 2\left(a\sum_{k=1}^{N}(x^k)^2 + \sum_{k=1}^{N} x^k(b-y^k)\right)$$

- si, on note $\overline{x}$ la moyenne (empirique) des x^k, $\overline{y}$ celle des y^k, $\overline{x^2}$ celle des $(x^k)^2$ et $\overline{xy}$ la moyenne du produit $x^k y^k$, on trouve (en annulant les dérivées) :

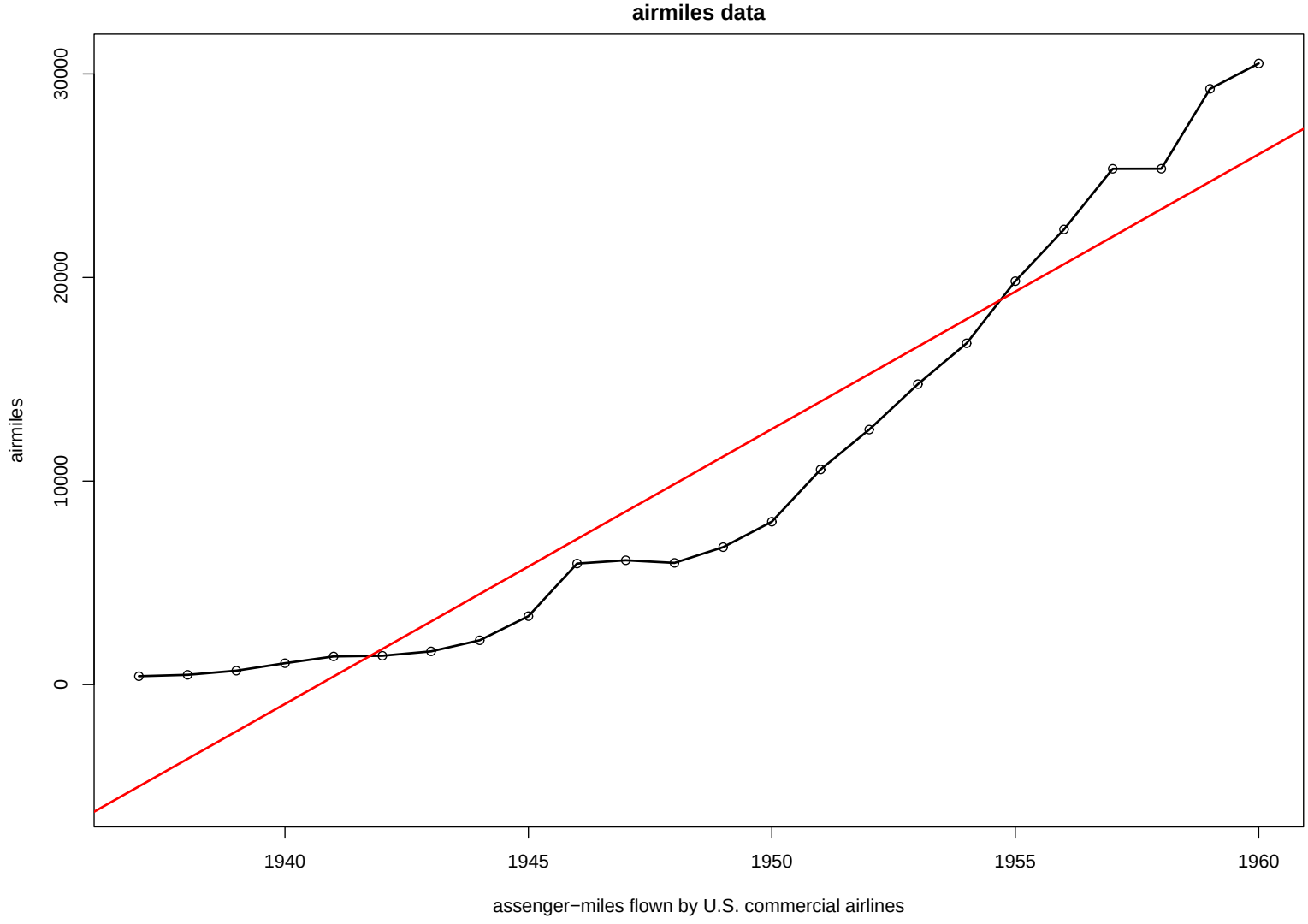
$$\begin{aligned}
\widehat{a} &= \frac{\overline{xy} - \overline{x}.\overline{y}}{\overline{x^2} - (\overline{x})^2} \\
\widehat{b} &= \frac{\overline{x^2}.\overline{y} - \overline{x}.\overline{xy}}{\overline{x^2} - (\overline{x})^2}
\end{aligned}$$

Exemple de mise en œuvre



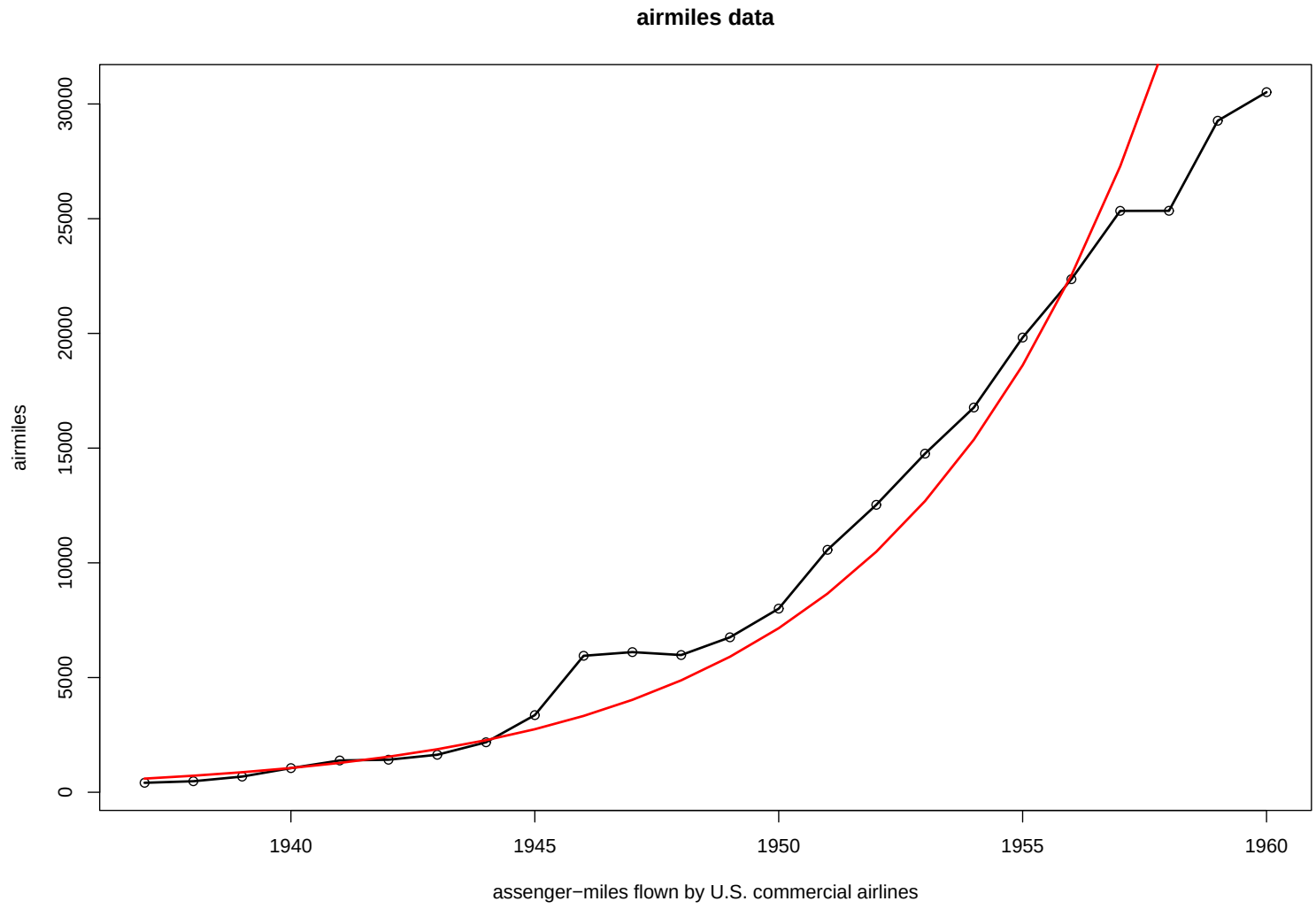
Données réelles : taille et poids de femmes américaines en 1975 (McNeil, D. R., 1977)

Ca ne fonctionne pas toujours ...



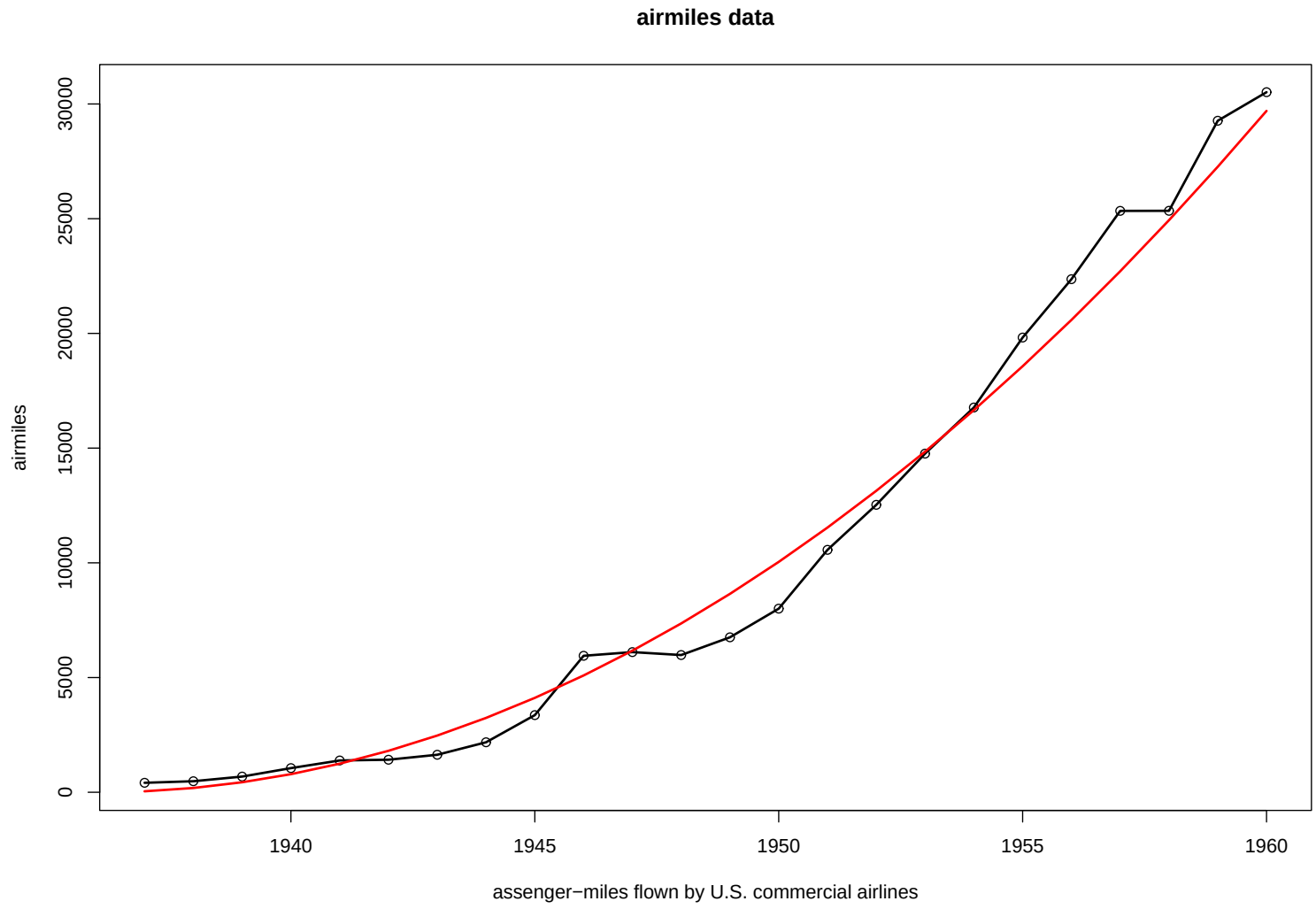
Données réelles : nombres de milles parcourus sur les compagnies US entre 1937 et 1960 (Brown, R. G., 1963)

Une petite transformation ...



Modèle linéaire sur $\log(airmiles)$!

Une autre transformation ...



Modèle linéaire sur $\sqrt{airmiles}$!

Modélisation statistique

On modélise le problème de la façon suivante :

- les $(x^k)_{1\leq k\leq N}$ sont donnés et non aléatoires
- y^k est une réalisation de la v.a. $Y^k = ax^k + b + U^k$
- les $(U^k)_{1\leq k\leq N}$ sont des v.a. indépendantes identiquement distribuées avec :
  - $E(U^k) = 0$
  - $\sigma^2(U^k) = \sigma_u^2$

Dans ce cadre, on montre que

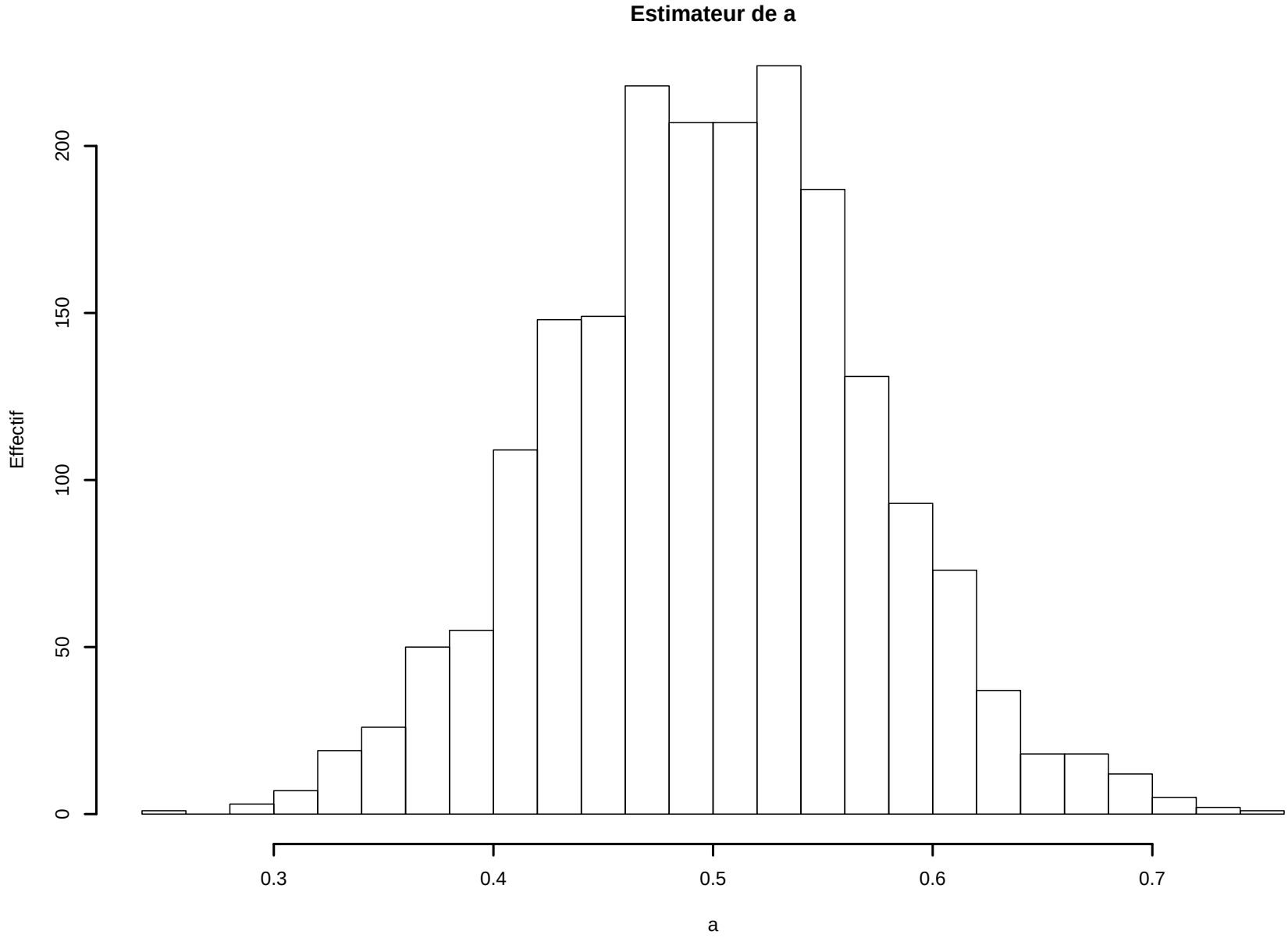
- $\widehat{a}$ et $\widehat{b}$ sont des estimateurs sans biais de a et b
- $\frac{1}{N-2}\sum_{k=1}^{N}\left(\widehat{a}x^k + \widehat{b} - y^k\right)^2$ est un estimateur sans biais de σ_u^2
- si les U^k sont des v.a. gaussiennes, on peut calculer de façon exacte la distribution de $(\widehat{a}, \widehat{b})$, ce qui permet de donner des intervalles de confiance pour les coefficients

Exemple

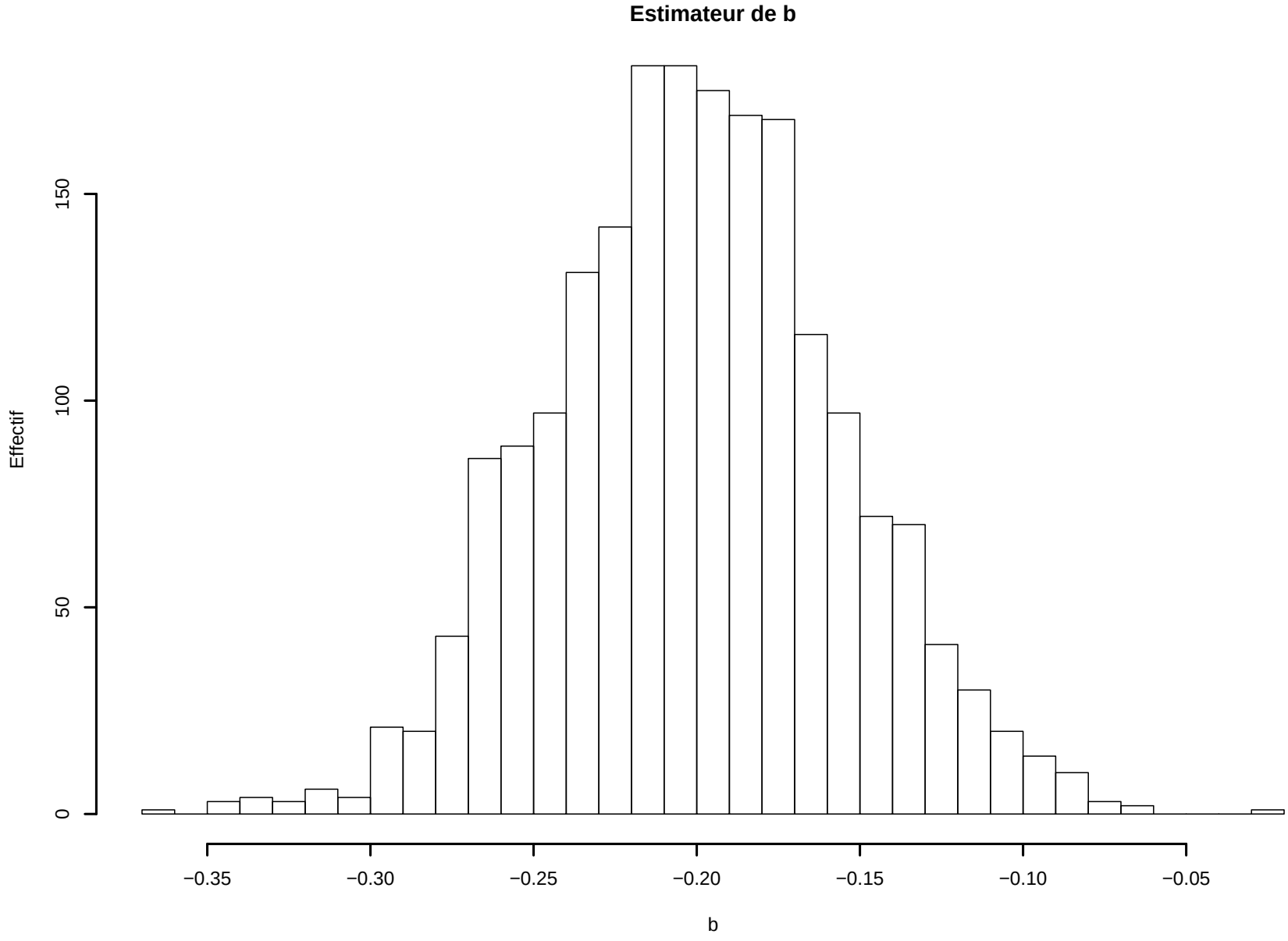
Protocole :

- on prend x^k équirépartis dans $[-1, 1]$ avec $N = 20$
- on engendre les U^k selon une loi normale d'écart-type 0.2
- on pose $Y^k = 0.5x^k - 0.2 + U^k$
- on calcule $\widehat{a}$ et $\widehat{b}$
- on recommence 2000 fois
- pour chaque estimateur, on trace l'histogramme des valeurs obtenues
- on calcule la moyenne des valeurs (approximation de l'espérance d'après la loi des grands nombres) :
  - $E(\widehat{a}) \simeq 0.5002$
  - $E(\widehat{b}) \simeq -0.2004$

Exemple (2)



Exemple (3)


Estimateur de b
Effectif
0
50
100
150
−0.35
−0.30
−0.25
−0.20
−0.15
−0.10
−0.05
b

Maximum de vraisemblance

Si on choisit un modèle pour les U^k, on peut procéder par maximum de vraisemblance :

- $U^k \sim \mathcal{N}(0, \sigma_u)$
- vraisemblance d'une observation

$$\mathcal{L}(a, b, x, y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-ax-b)^2}{2\sigma^2}}$$

- vraisemblance des N observations

$$\mathcal{L}(a, b, (x^k), (y^k)) = \frac{1}{(2\sigma_u\pi)^{\frac{N}{2}}} e^{-\frac{\sum_{k=1}^{N}(y_k-ax_k-b)^2}{2\sigma_u^2}}$$

- Maximiser la vraisemblance revient à maximiser la log-vraisemblance, soit ici minimiser $\sum_{k=1}^{N}(y_k - ax_k - b)^2$!
- Bruit gaussien $\Rightarrow$ MV équivalent aux moindres carrés

Résumé des propriétés théoriques

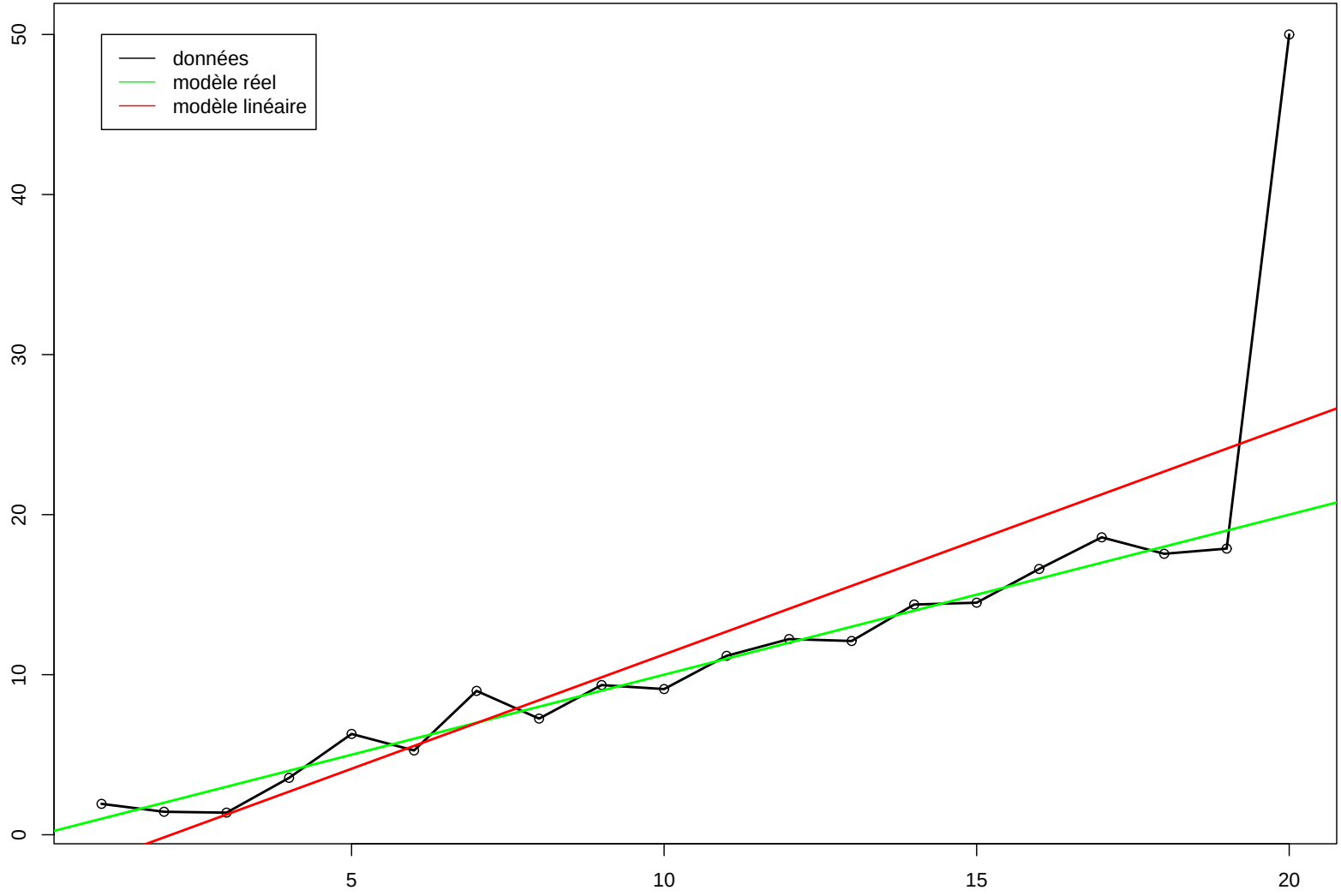
On peut généraliser le modèle au cas où les x^k sont des réalisations de v.a. X^k. On fait donc les hypothèses suivantes :

- X et Y sont deux v.a. telles que $E(Y|X=x) = ax + b$
- $\sigma^2(Y|X=x) = \sigma_u^2$
- les (x^k, y^k) sont des réalisations indépendantes de (X, Y)

Dans ce cas :

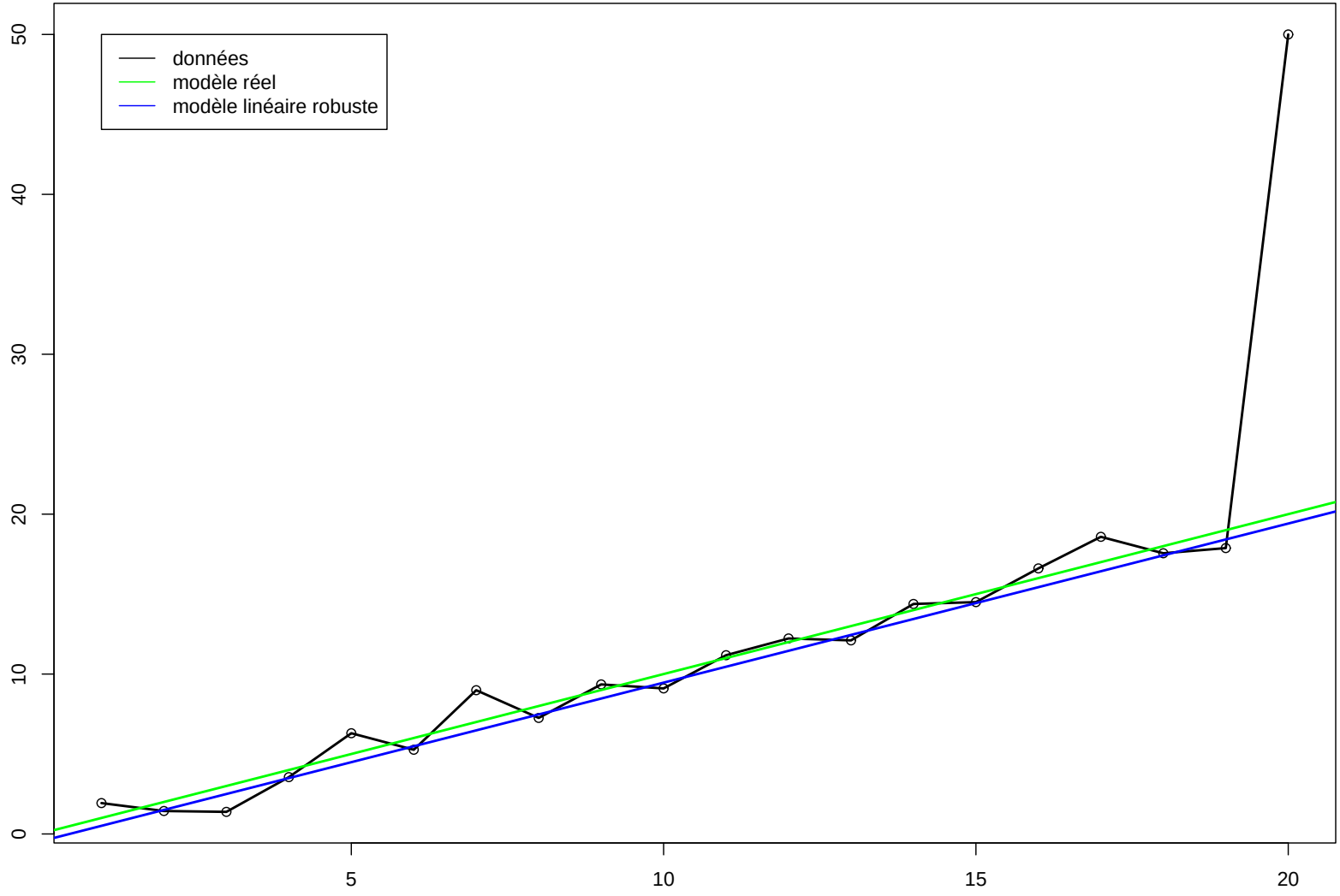
- les estimateurs de a et b obtenus par le critère des moindres carrés (MC) sont sans biais
- l'erreur résiduelle divisée par $N-2$ est un estimateur sans biais de σ_u^2
- si $Y - E(Y|X) \sim \mathcal{N}(0, \sigma_u)$:
  - les estimateurs MC sont ceux du maximum de vraisemblance
  - on peut calculer explicitement la distribution de (a, b)

Attention : robustesse des MC



Moindres carrés classiques

Attention : robustesse des MC



Moindres carrés robustes

Cas général : la régression linéaire multiple

Dans le cas général, x^k et y^k sont des vecteurs. La méthode des moindres carrés conduit à minimiser

$$\widehat{\mathcal{E}}(A,b) = \sum_{k=1}^{N} \left\| Ax^k + b - y^k \right\|^2$$

Pour la résolution, on écrit :

$$Ax + b = Cz,$$

avec

$$\begin{aligned} C &= (Ab) \\ z &= \begin{pmatrix} x \\ 1 \end{pmatrix} \end{aligned}$$

Régression linéaire multiple (2)

On montre que la différentielle de $\mathcal{E}(C)$ s'écrit :

$$2\sum_{k=1}^{N}\left(\left(Cz^k - y^k\right) z^{kT}\right)$$

En posant

$$\begin{aligned} Z &= (z^1 z^2 \dots z^N) \\ Y &= (y^1 y^2 \dots y^N) \end{aligned}$$

la différentielle devient :

$$2(CZ - Y)Z^T$$

Régression linéaire multiple (3)

On doit donc résoudre :

$$ZZ^T C^T = ZY^T$$

Deux cas :

1. on peut calculer l'inverse de ZZ^T (algorithme en $O(n^3)$)
2. ZZ^T n'est pas inversible en pratique : on utilise une SVD (décomposition en valeurs singulières, algorithme en $O(n^4)$)

Il y a toujours une solution et on sait toujours la calculer en pratique. Cette solution, $\widehat{C}$, est l'estimateur des moindres carrés du modèle linéaire.

Singular Value Decomposition

Soit une matrice A, $m \times n$. La SVD de A est un triplet de matrices, U, D et V telles que :

- D est une matrice $n \times n$ diagonale dont les termes diagonaux (les valeurs singulières) sont positifs ou nuls et rangées par ordre décroissant
- U et V sont des matrices orthonormées
- $A = VDU^T$
- le nombre de valeurs singulières strictement positives donne le rang de A

Exemple :

$$\begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix} = \frac{\sqrt{5}}{5} \begin{pmatrix} -1 & -2 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} \sqrt{10} & 0 \\ 0 & 0 \end{pmatrix} \frac{\sqrt{2}}{2} \begin{pmatrix} -1 & -1 \\ -1 & 1 \end{pmatrix}$$

Singular Value Decomposition (2)

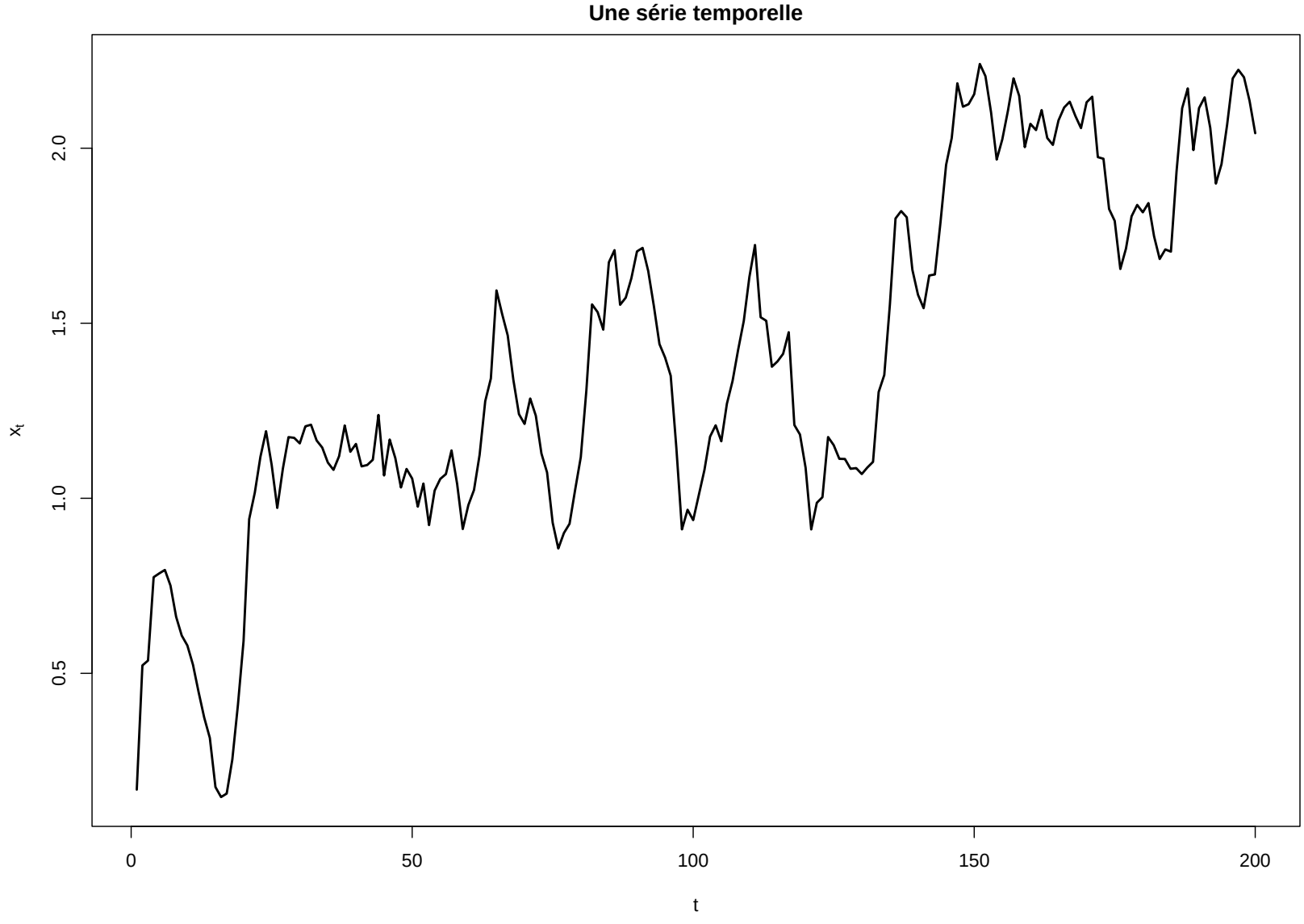
Permet de définir la pseudo-inverse d'une matrice :

- dans D, on remplace les petites valeurs singulières (par exemple $< 10^{-8}$) par 0, ce qui donne $\widetilde{D}$. Par exemple, s'il ne reste que 2 valeurs non nulles :

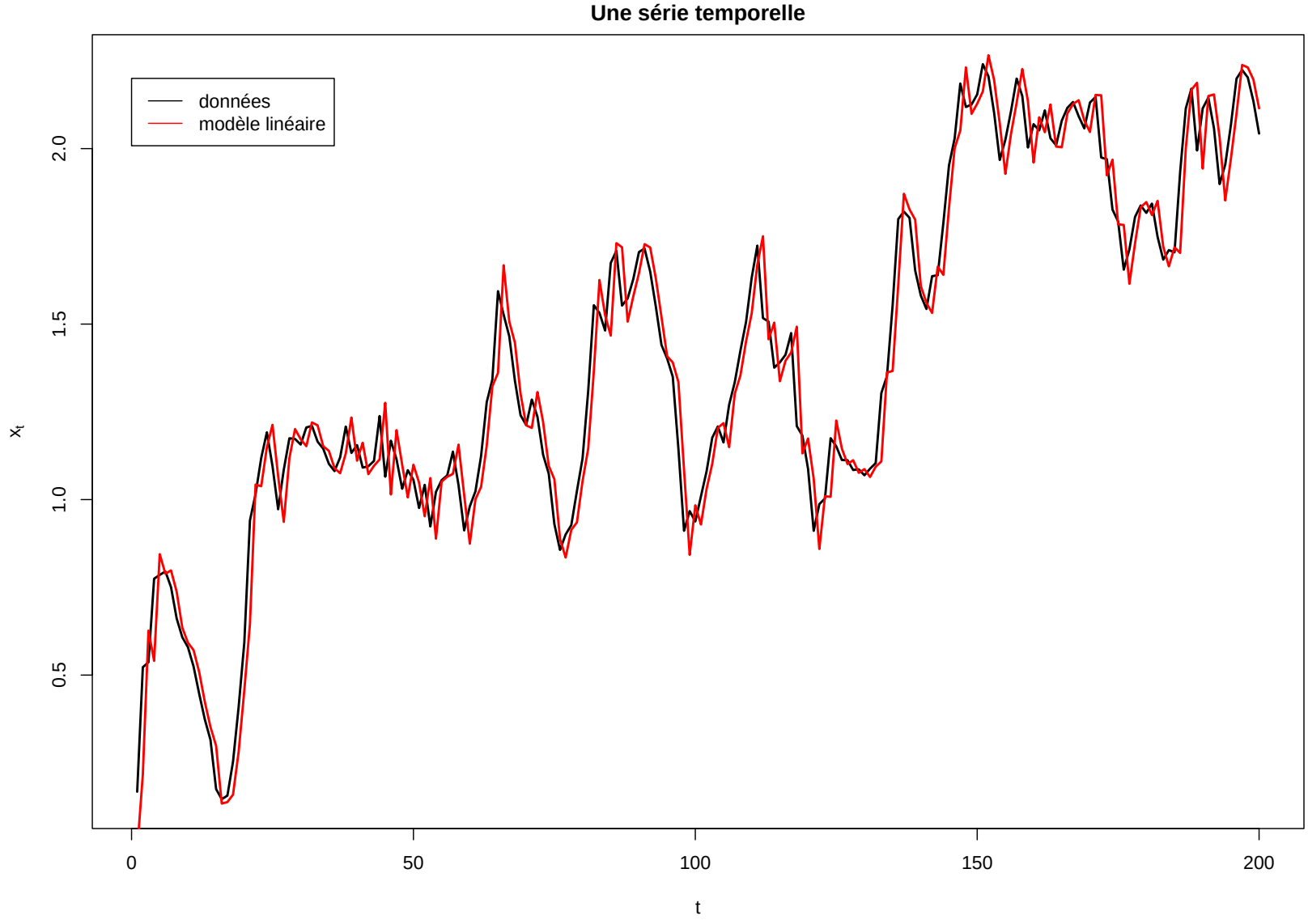
$$\widetilde{D} = \begin{pmatrix} \alpha_1 & 0 & 0 & \cdots & 0 \\ 0 & \alpha_2 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & 0 & \vdots & 0 & \vdots \\ 0 & \vdots & \vdots & \cdots & 0 \end{pmatrix}$$

- $A \simeq V \widetilde{D} U^T$
- on définit $D^\dagger$ comme la matrice diagonale dont les seuls termes diagnonaux non nuls sont les inverses des α_i
- on définit la pseudo-inverse de A par $A^\dagger = V D^\dagger U^T$

Exemple : série temporelle

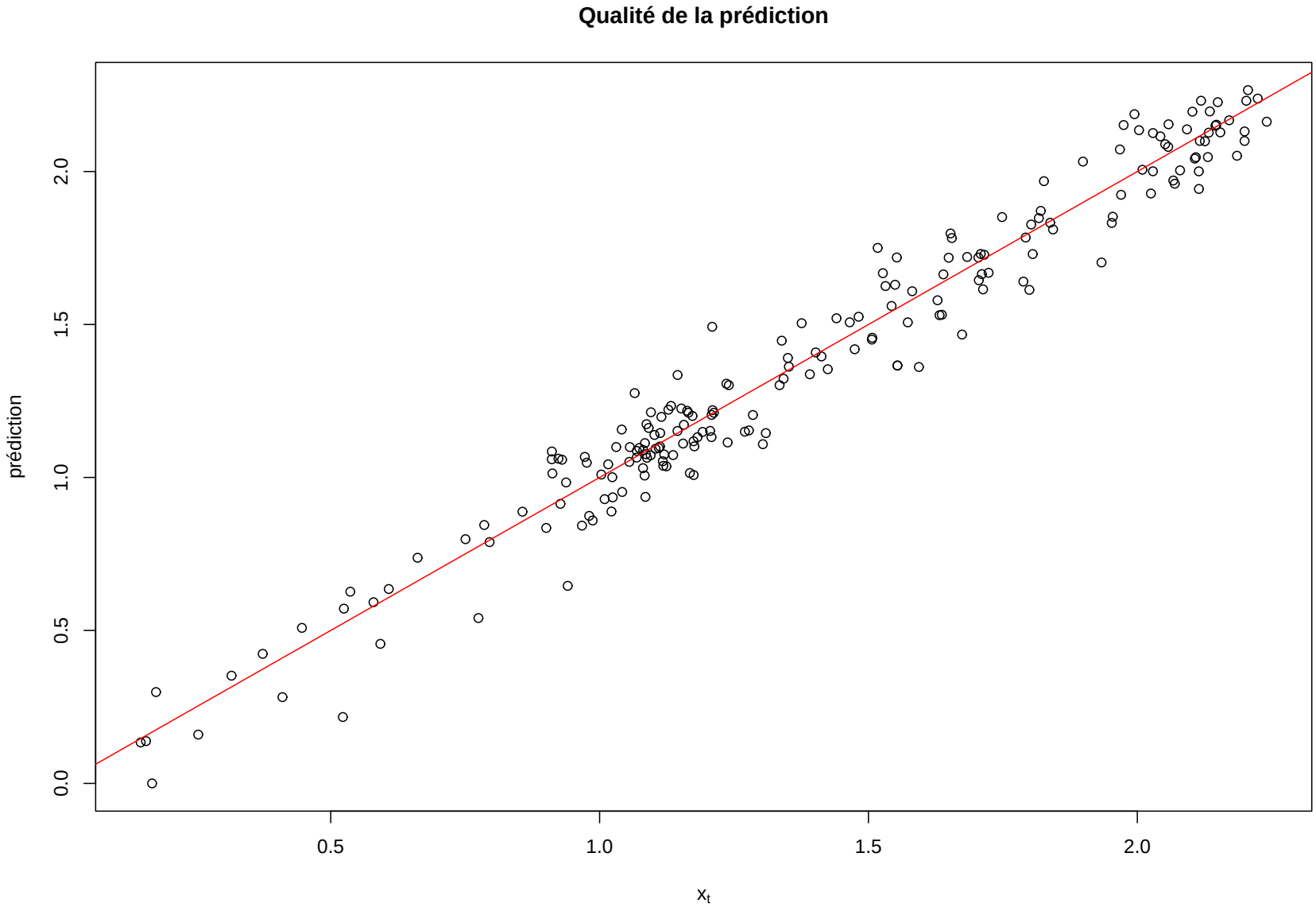


Exemple : série temporelle



Prédiction par modèle linéaire !

Exemple


Qualité de la prédiction
prédiction
0.0
0.5
1.0
1.5
2.0
0.5
1.0
1.5
2.0
x_t

Modélisation auto-regressive

On écrit

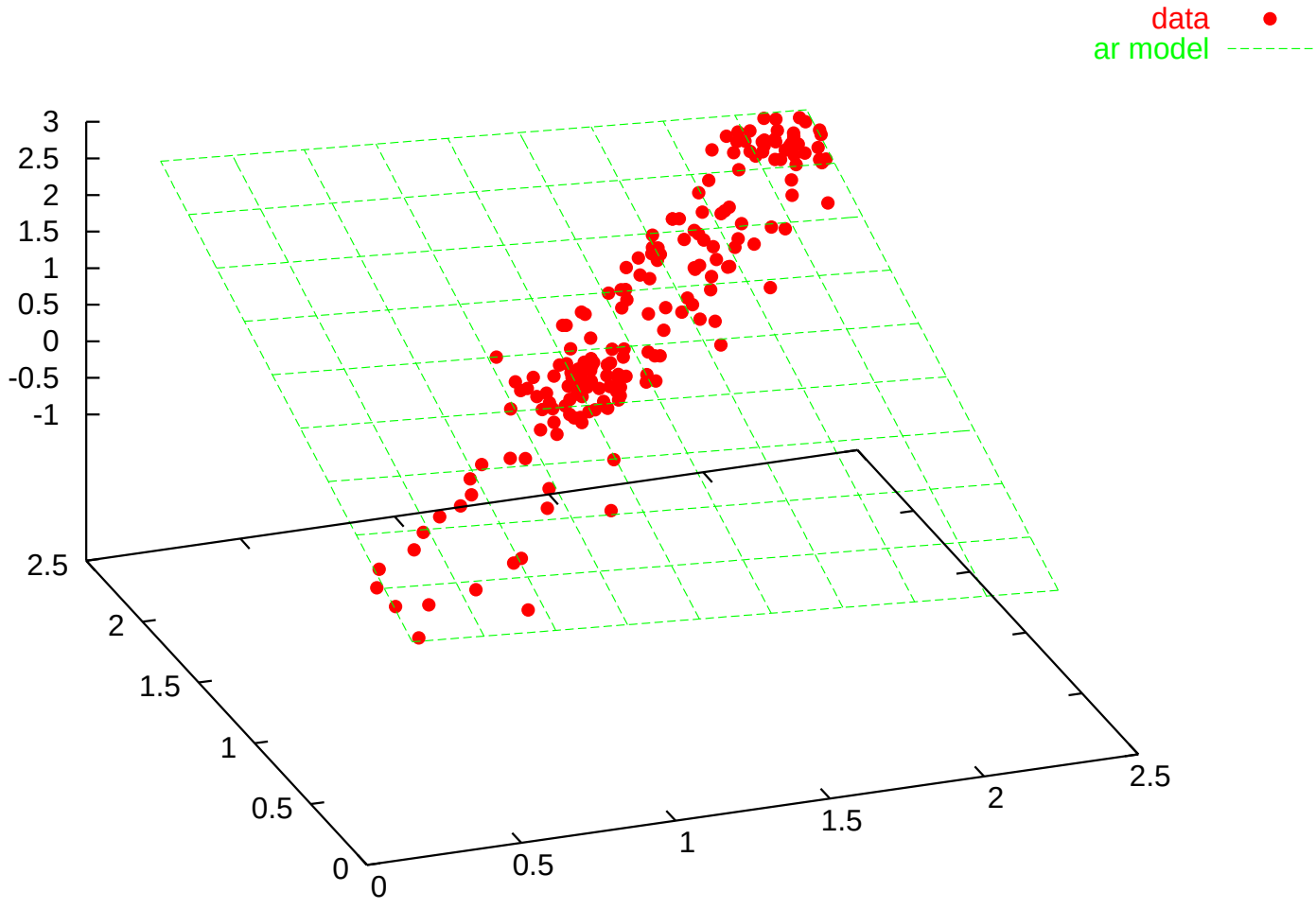
$$x_t = a_1 x_{t-1} + a_2 x_{t-2} + \ldots + a_n x_{t-n} + \epsilon(t)$$

On trouve les coefficients par moindres carrés (par exemple).

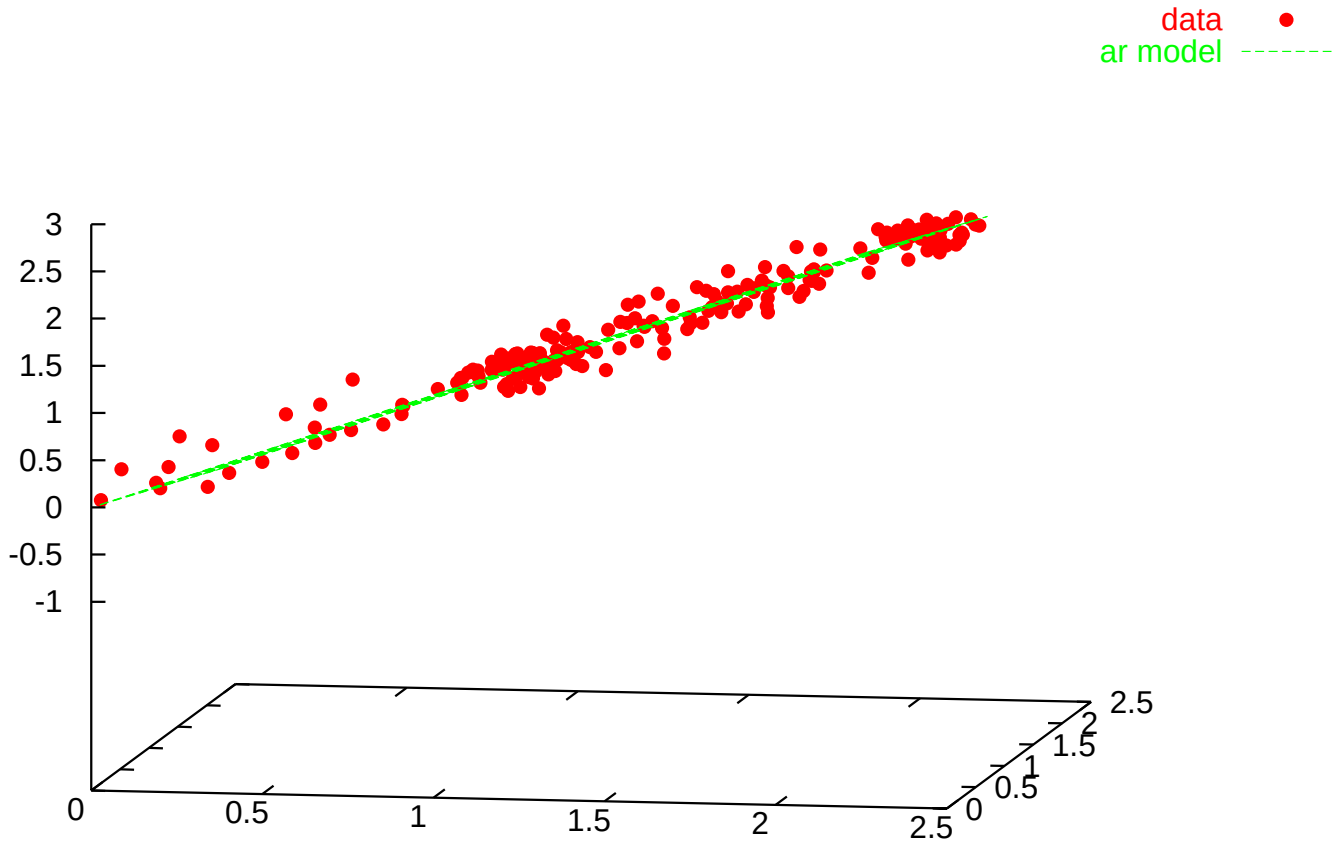
Grosse difficulté : choix de l'ordre du modèle (i.e., la valeur de n).

Dans notre exemple, $n = 2$.

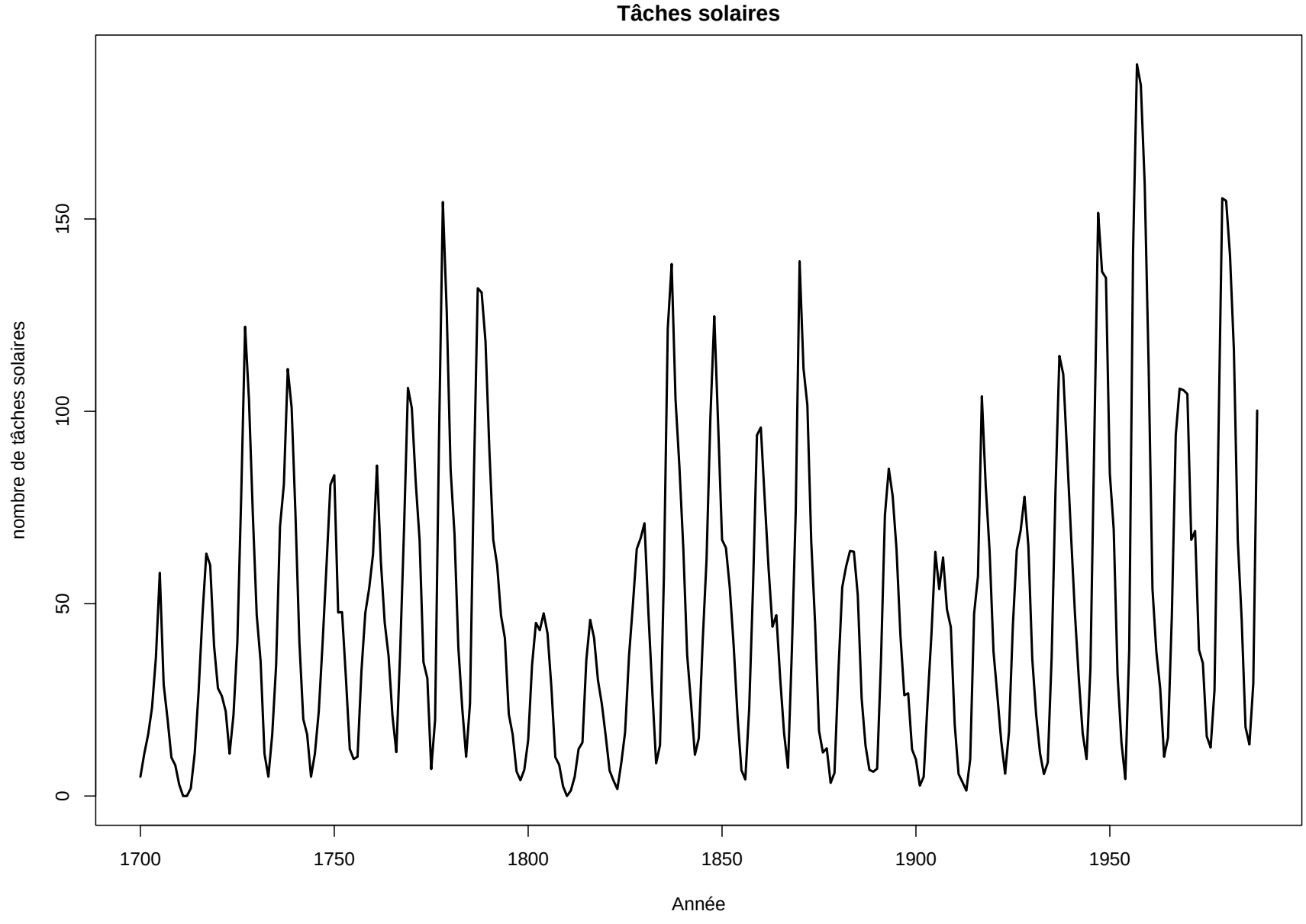
Représentation graphique en “3D”



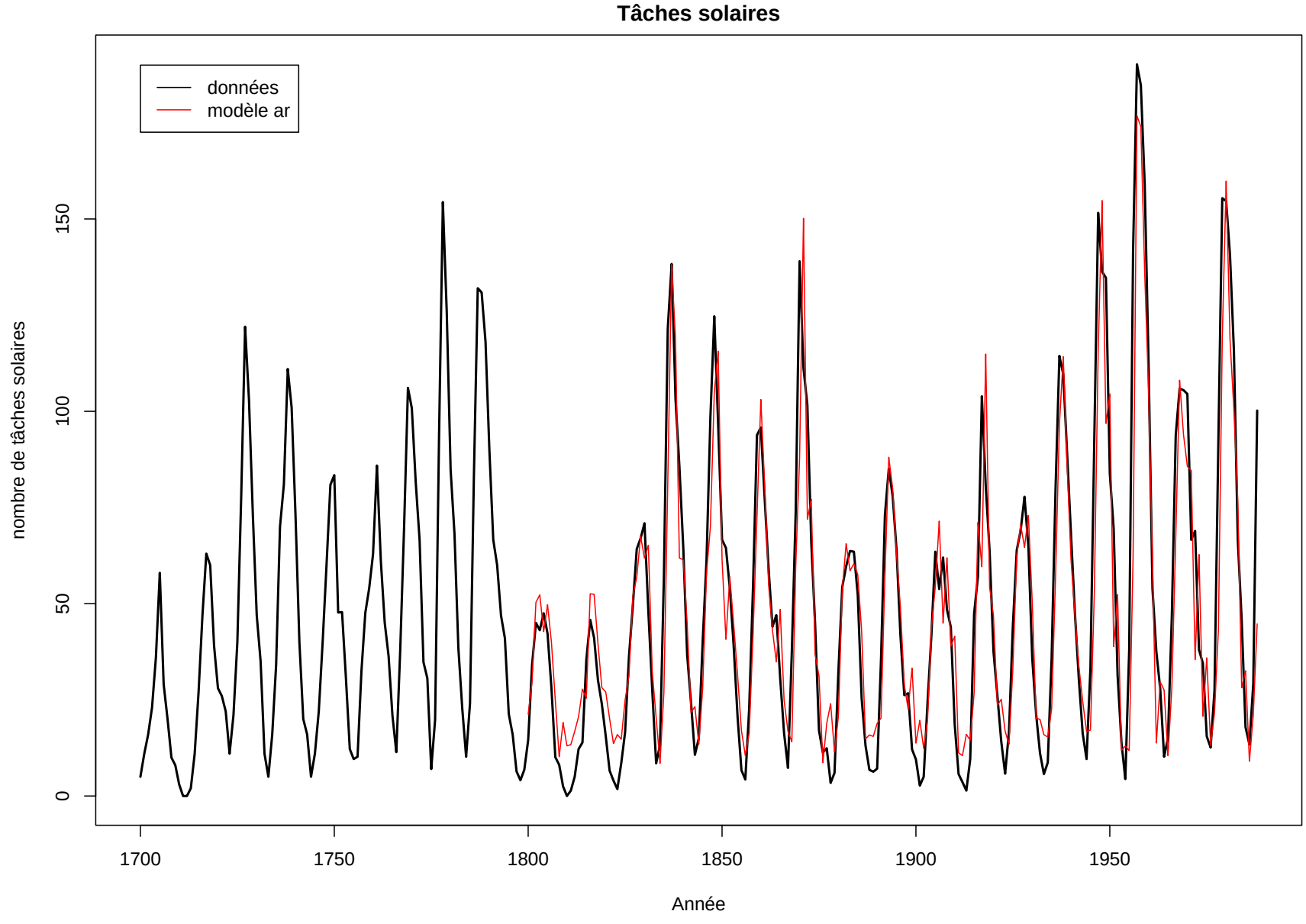
Représentation graphique en “3D”



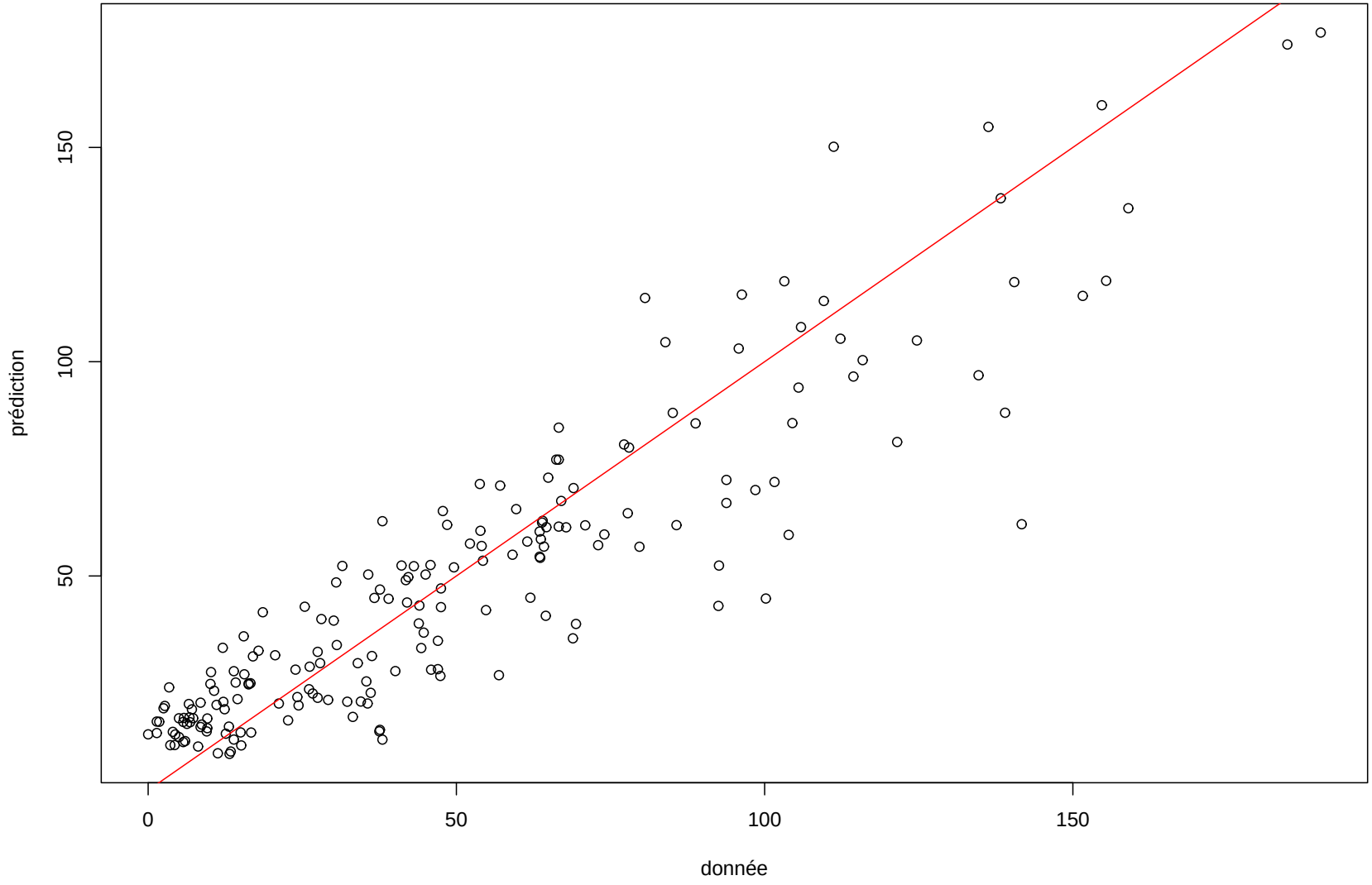
Données réelles



Données réelles



Qualité de la prédiction



Propriétés théoriques

Exactement les mêmes que pour la régression simple.
Hypothèses :

- X et Y sont deux v.a. telles que $E(Y|X=x) = Ax + b$
- $cov(Y|X=x) = \sigma_u^2 I$ (covariance)
- les (x^k, y^k) sont des réalisations indépendantes de (X, Y)

Dans ce cas :

- les estimateurs de A et b obtenus par le critère des moindres carrés (MC) sont sans biais
- l'erreur résiduelle divisée par $N - n - 1$ est un estimateur sans biais de σ_u^2
- si $Y - E(Y|X) \sim \mathcal{N}(0, \sigma_u I)$:
  - les estimateurs MC sont ceux du maximum de vraisemblance
  - on peut calculer explicitement la distribution de (A, b)

Résumé

En résumé, **si les données suivent un modèle linéaire** :

- on peut estimer simplement ses paramètres (algèbre linéaire)
- l'estimation est sans biais
- elle correspond au maximum de vraisemblance pour une erreur gaussienne
- on peut donner des intervalles de confiance, etc.

Problème : que se passe-t-il si les données **ne suivent pas un modèle linéaire** ?

Régression non linéaire

La régression linéaire est un cas particulier du modèle de régression **paramétrique** :

- on suppose donnée une fonction F de $W \times \mathbb{R}^n$ dans $\mathbb{R}^p$
- W les paramètres (numériques) du modèle
- on cherche $w \in W$ tel que $F(w, x^k) \simeq y^k$

Exemple :

- le modèle linéaire pur, $F(A, x^k) = Ax^k$, où A est une matrice $p \times n$
- le modèle affine, $F((A, b), x^k) = Ax^k + b$, où A est une matrice $p \times n$ et b un vecteur de $\mathbb{R}^p$
- les modèles linéaires généralisés et les perceptrons multi-couches (cf prochains cours)

Problème : comment choisir w ?

Une solution : les moindres carrés !

Pourquoi les moindres carrés ?

Modélisation probabiliste : (x^k, y^k) réalisations d'une suite de variables aléatoires i.i.d. (X^k, Y^k).
Loi forte des grands nombres :

$$\widehat{\mathcal{E}}_N(w) = \frac{1}{N} \sum_{k=1}^{N} \left\| F(w, x^k) - y^k \right\|^2 \to E\left(\|F(w, X) - Y\|^2 \right) \ p.s.$$

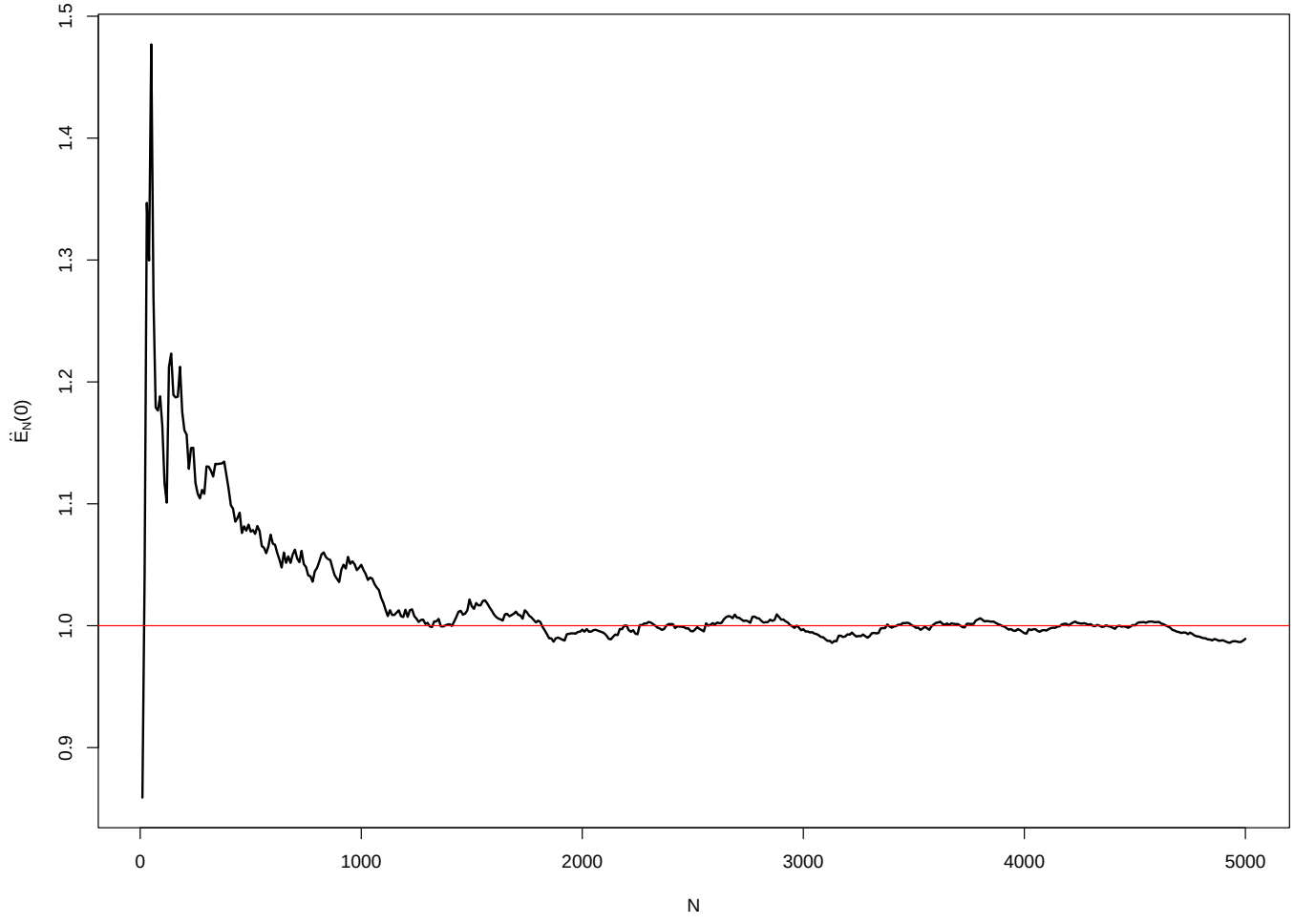
En fait, la convergence est **uniforme** et on a :

$$\lim_{N \to \infty} d(w_N, W) = 0 \ p.s.$$

avec w_N un minimiseur de $\widehat{\mathcal{E}}_N(w)$ et W l'ensemble des minima de $E\left(\|F(w, X) - Y\|^2 \right)$

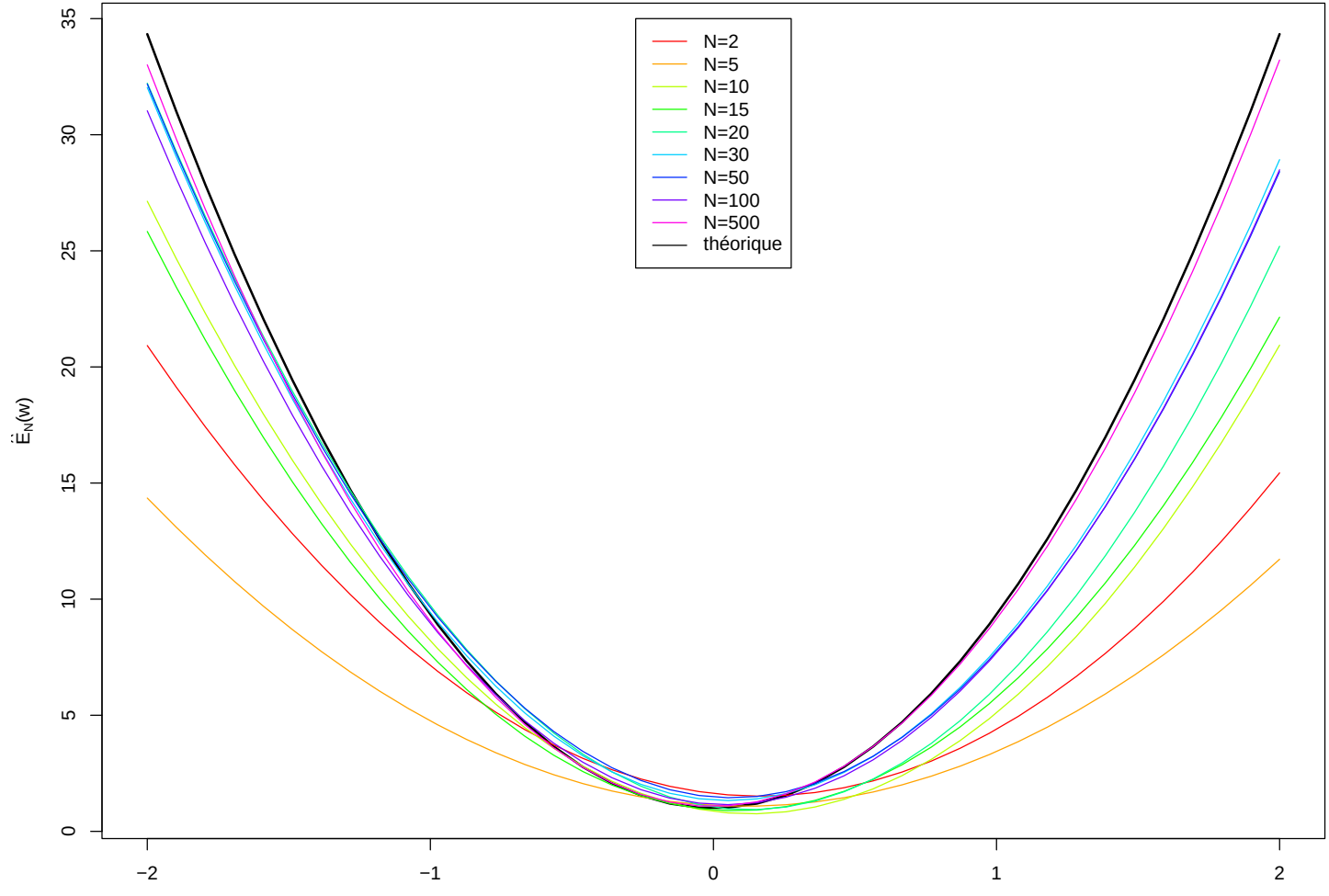
Exemple

On prend $y^k = e^k$, un bruit gaussien de variance 1. Modèle $y^k = wx^k$. Pour $w = 0$, on obtient l'erreur quadratique suivante :

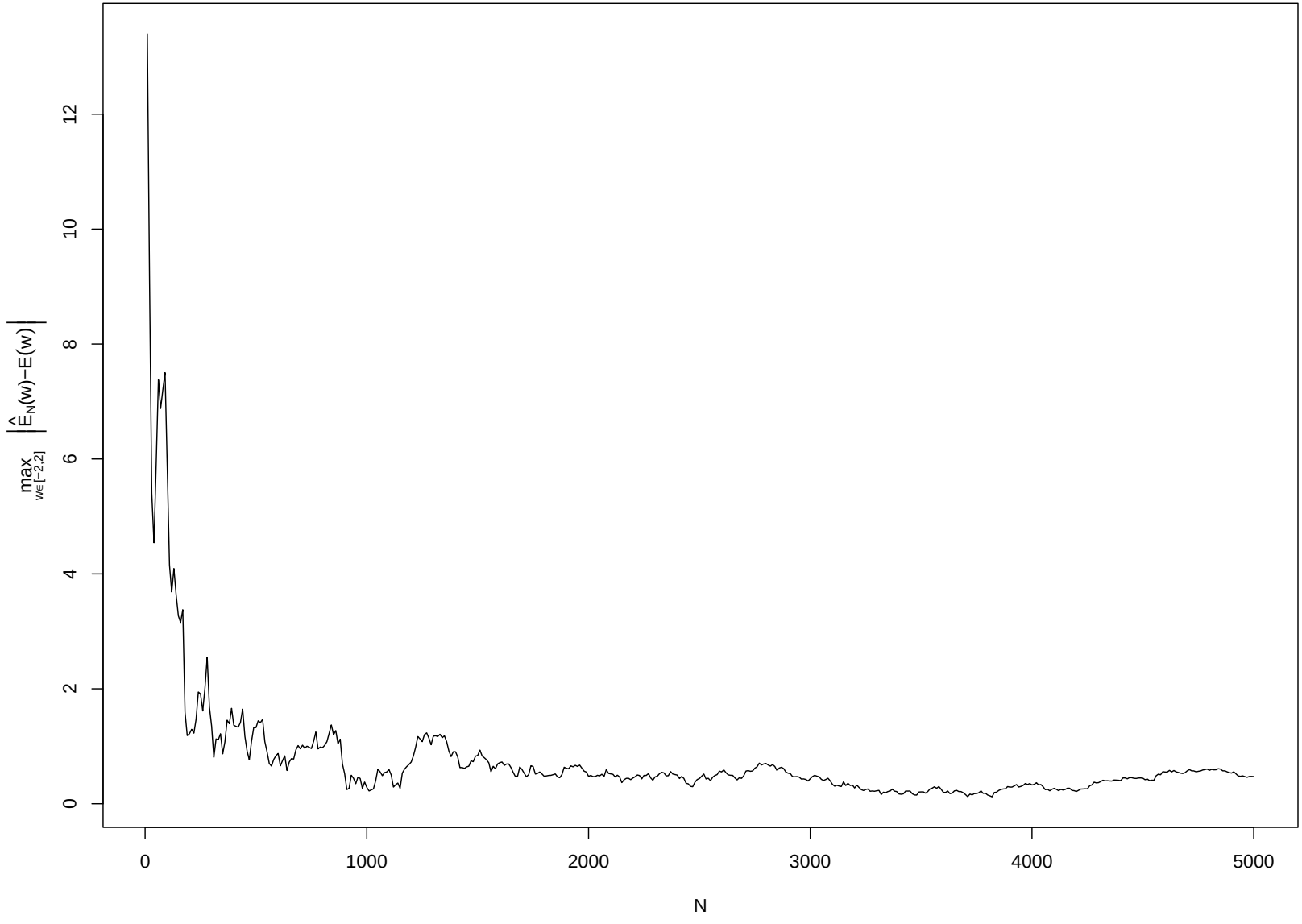


Exemple (2)

On considère l'erreur du modèle pour w dans $[-2, 2]$, avec entre 2 et 500 observations. Les x^k sont choisis aléatoirement uniformément dans $[-5, 5]$. Erreur théorique : $1 + \frac{25}{3}w^2$.



Exemple (3)



Ecarts maximaux (en valeur absolue) à l'erreur théorique

Pourquoi les moindres carrés ? (2)

Par définition :

$$\lambda(w) = E\left(\|F(w,X) - Y\|^2\right) = \int \|F(w,x) - y\|^2 \, p(x,y) dx dy$$

On conditionne, i.e., on utilise $p(x,y) = p(y|x)p(x)$. Après quelques manipulations, on obtient :

$$\lambda(w) = \int \|F(w,x) - E(Y|x)\|^2 \, p(x) dx + \int \left(E(\|Y\|^2|x) - \|E(Y|x)\|^2\right) p(x) dx$$

En minimisant λ, on approche donc $E(Y|x)$. L'estimateur de w par les moindres carrés donne donc asymptotiquement la meilleure approximation de $E(Y|x)$ par $F(w,x)$.

Maximum de vraisemblance

On reprend le raisonnement du cas linéaire :

- on suppose que le modèle s'écrit $Y^k = F(w, X^k) + U^k$
- U^k est un vecteur gaussien centré de matrice de covariance $\sigma_u I$
- la vraisemblance de N observations s'écrit alors

$$\mathcal{L}(w, (x^k), (y^k)) = \frac{1}{(2\sigma_u \pi)^{\frac{N}{2}}} e^{-\frac{\sum_{k=1}^{N} (y_k - F(w, x_k))^2}{2\sigma_u^2}}$$

- comme pour le modèle linéaire, maximiser la vraisemblance revient donc à minimiser l'erreur quadratique

Régression linéaire : résumé

Réseau à une couche avec neurones linéaires :

- permet de modéliser une relation affine $y = Ax + b$
- les moindres carrés correspondent à la recherche d'une approximation de $E(y|x)$
- les moindres carrés sont équivalents au maximum de vraisemblance dans le cas d'un bruit gaussien
- les coefficients optimaux sont toujours calculables
- le modèle est simple mais utile

A toujours utiliser comme référence avant un traitement par un réseau de neurones plus puissant.

Application à la discrimination

Il suffit de construire y^k en fonction de la classe de x^k (**codage disjonctif complet**) :

- problème à deux classes : $x^k \in C_1 \Rightarrow y^k = 1$, $x^k \in C_2 \Rightarrow y^k = 0$
- problème à q classes : $y^k \in \mathbb{R}^q$, avec $x^k \in C_j \Leftrightarrow y^k_q = \delta_{qj}$

  Par exemple pour trois classes, la classe 2 correspond à $(0, 1, 0)$.

  Attention ! Ne jamais utiliser un codage numérique (par exemple $x^k \in C_j \Rightarrow y^k = j$), cela induit une structure artificielle sur la fonction à modéliser.

Cas linéaire : même méthode de résolution (moindres carrés).

Classement d'un nouvel individu

Comment classer x ? Solution intuitive :

- deux classes : si $F(w, x) > 0.5 \Rightarrow C_1$, sinon C_2
- q classes : on calcule $y = F(w, x)$ et on affecte x à C_j tel que $y_j = \max_l y_l$.

Pourquoi cela fonctionne-t-il ?

$F(w, x)$ approche $E(Y|x)$:

- deux classes : $E(Y|X = x) = P(C_1|x)$
- q classes : $E(Y_j|X = x) = P(C_j|x)$

Donc, $F(w, x) > 0.5$ correspond approximativement à $P(C_1|x) > 0.5$.

$\Rightarrow$ l'affectation à C_1 est la décision optimale.

Classement optimal

On coupe l'espace $\mathbb{R}^n$ en q zones (quelconques), $\mathcal{R}_j$.
Probabilité de ne pas faire d'erreur en classant x :

$$P(ok) = \sum_{j=1}^{q} P(x \in \mathcal{R}_j, C_j)$$

On a

$$\begin{aligned} P(ok) &= \sum_{j=1}^{q} P(x \in \mathcal{R}_j | C_j) P(C_j) \\ &= \sum_{j=1}^{q} \int_{\mathcal{R}_j} p(x|C_j) P(C_j) dx \end{aligned}$$

Classement optimal (2)

Pour maximiser $P(ok)$, il suffit de définir $\mathcal{R}_j$ par :

$$\mathcal{R}_j = \{x \mid p(x|C_j)P(C_j) > p(x|C_t)P(C_t) \text{ pour } t \neq j\}$$

Or

$$p(x|C_j)P(C_j) = P(C_j|x)p(x)$$

Il suffit donc de choisir j qui maximise $P(C_j|x)$.
Extension : notion de risque. L_{kj} coût pour la décision k alors que la vraie classe est j. Coût total :

$$R = \sum_{j=1}^{q} \int_{\mathcal{R}_j} \sum_{k=1}^{q} L_{kj} p(x|C_k) P(C_k) dx$$

Application : diagnostic médical par exemple.

Le cas gaussien

Distribution gaussienne en dimension quelconque (n) :

$$p(x) = \frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{|\Sigma|}} \exp^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

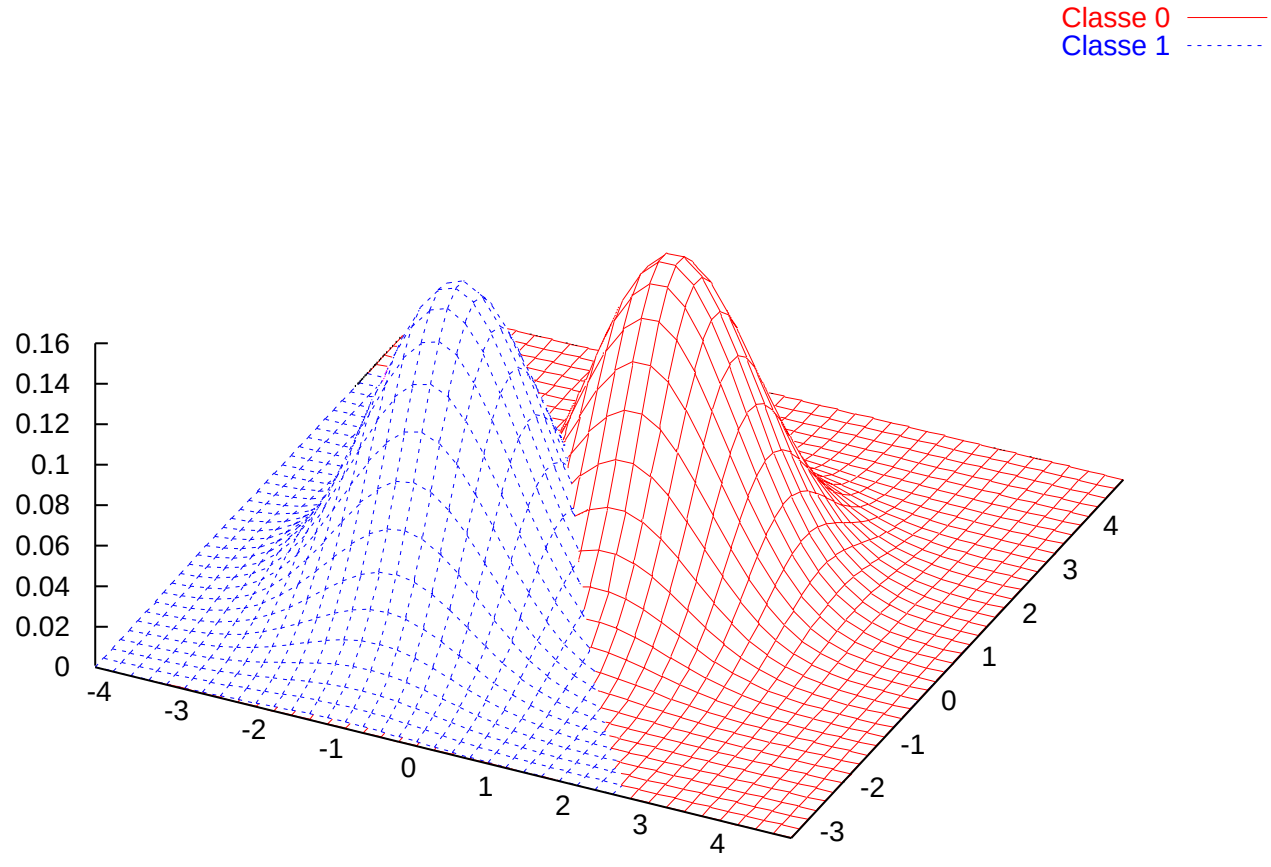
μ est le vecteur moyenne et Σ la matrice de covariance ($|\Sigma|$ est le déterminant de la matrice). On a :

$$\begin{aligned} E(x) &= \mu \\ E\left((x-\mu)(x-\mu)^T\right) &= \Sigma \end{aligned}$$

Mélange de q gaussiennes :

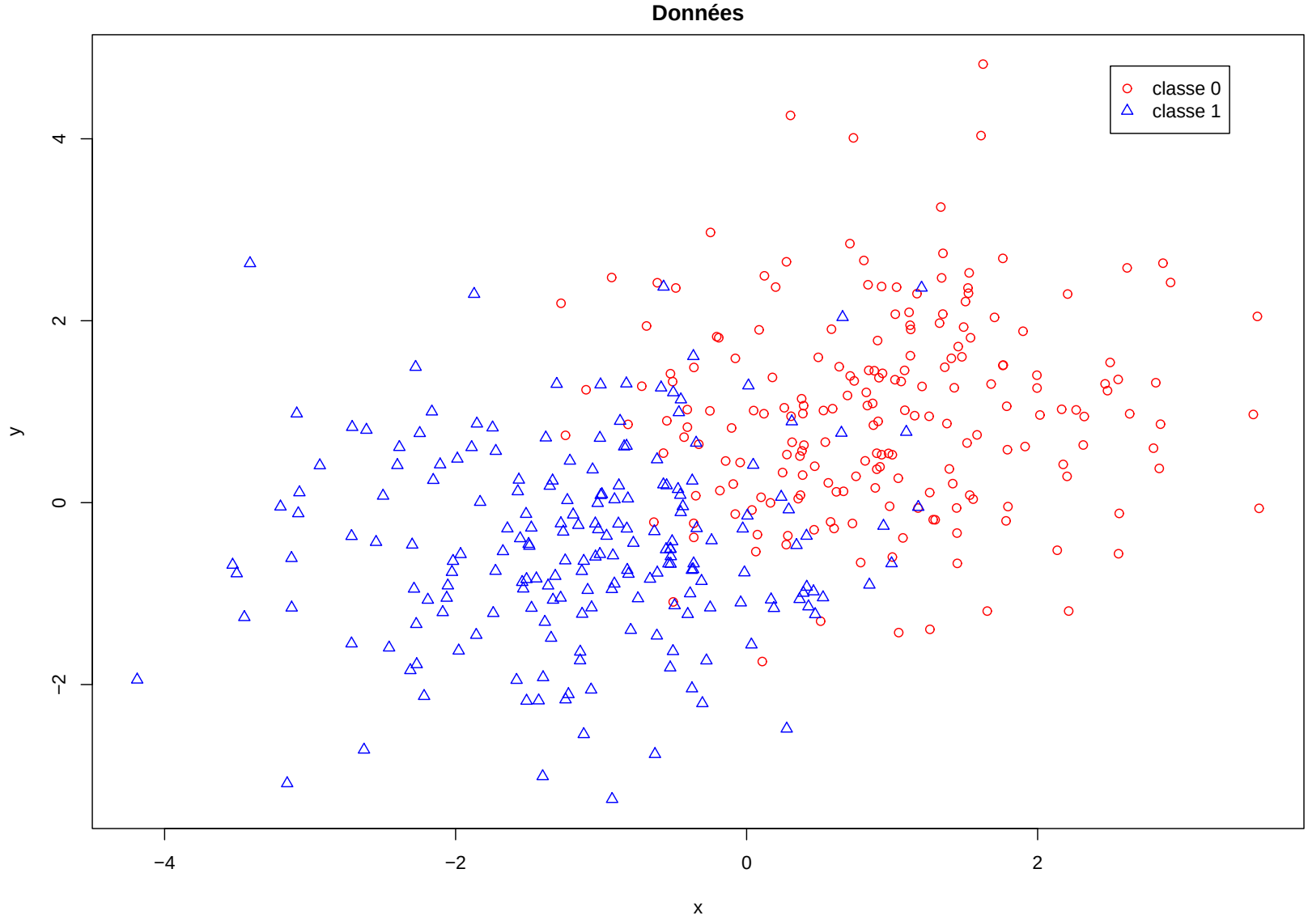
$$p(x) = \sum_{j=1}^{q} P(C_j)p(x|C_j)$$

Exemple



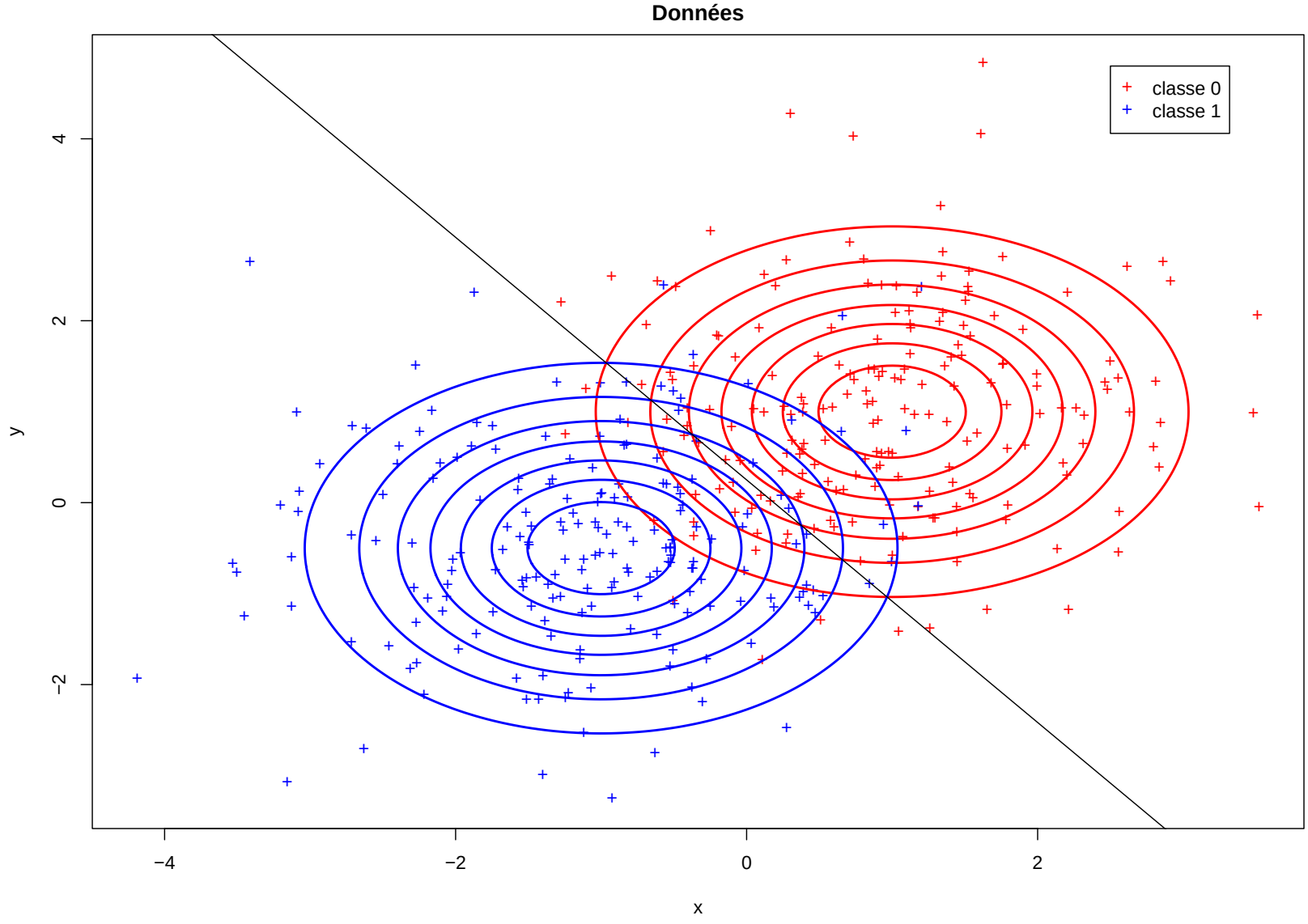
2 gaussiennes

Exemple (simulation)



2 gaussiennes

Exemple (simulation)



2 gaussiennes

Classement optimal gaussien

Maximiser $p(x|C_j)P(C_j)$ revient à maximiser $\ln(p(x|C_j)P(C_j))$, c'est-à-dire :

$$d_j(x) = -\frac{1}{2}(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j) - \frac{1}{2}\ln|\Sigma_j| + \ln P(C_j)$$

Pour chaque classe on a une forme quadratique en x : frontières quadratiques.
Si on a une matrice Σ unique, trouver le $d_j(x)$ maximum revient à trouver le $f_j(x)$ maximum :

$$f_j(x) = \mu_j^T\Sigma^{-1}x - \frac{1}{2}\mu_j^T\Sigma^{-1}\mu_j + \ln P(C_j)$$

Les f_j sont linéaires : frontières linéaires.

Maximum de vraisemblance

Le raisonnement appliqué à la régression n'a plus de sens :

- cible 1 ou 0
- bruit gaussien : aucun sens

Pour un problème à deux classes, on veut que $F(w, x^k)$ approche la probabilité $P(C_1|x^k)$. La vraisemblance peut alors s'écrire :

$$p(y^k|w) = \prod_{k=1}^{N} F(w, x^k)^{y^k} (1 - F(w, x^k))^{1-y^k}$$

On minimise l'*entropie croisée* (problème difficile !) :

$$E(w) = -\sum_{k=1}^{N} \left(y^k \ln F(w, x^k) + (1 - y^k) \ln(1 - F(w, x^k)) \right)$$

Fonctions discriminantes

Le modèle linéaire donne en général une mauvaise approximation de $P(C_j|x)$, même dans le cas où il permet un classement optimal.

Pour deux classes, la décision optimale est 1 si $P(C_1|x) > P(C_2|x)$, 2 sinon. Il suffit donc de connaître le signe de $\phi(P(C_1|x)) - \phi(P(C_2|x))$, où ϕ est une fonction croissante.

De façon générale, il suffit d'obtenir une fonction discriminante $\psi(x)$ telle que $\psi(x) > 0$ si et seulement si $P(C_1|x) > P(C_2|x)$.

Activation non linéaire

Il est courant d'utiliser pour T une fonction croissante non linéaire.
Exemple principal, la sigmoïde logistique :

$$T(x) = \frac{1}{1 + e^{-x}}$$

Motivation : approximation de $P(C_1|x)$ pour deux classes gaussiennes.

Inconvénient : on ne peut plus calculer les paramètres optimaux $\Rightarrow$ optimisation de $\mathcal{E}(w)$ par descente de gradient.

Activation non linéaire (2)

Historiquement, on a beaucoup utilisé la fonction *step* de Heaviside :

$$T(x) = \begin{cases} 0 & \text{quand } x < 0 \\ 1 & \text{quand } x \geq 0 \end{cases}$$

- “Modèle” des neurones biologiques (McCulloch et Pitts, 1943)
- perceptron (Rosenblatt, 1962), avec souvent 0 remplacé par -1
- adaline (Widrow et Hoff, 1960)

Discrimination linéaire

Comme la décision se fait toujours par une règle de la forme $F(w, x) > \alpha$ et T est croissante, la décision reste linéaire et donc limitée.

Le plus simple problème non linéairement séparable en dimension 2 se généralise en dimension n, avec $n + 2$ points.

Le modèle linéaire est limité, mais adapté à certains problèmes.

Le perceptron

En première approche : un réseau de neurones à une couche dont les neurones utilisent la fonction de Heaviside modifiée ($T(x) = -1$ quand $x < 0$, 1 sinon).
Principal problème : l'apprentissage (l'estimation des paramètres optimaux) car on souhaite minimiser le nombre de mauvais classements.
Critère du perceptron pour une sortie :

$$\mathcal{E}(a) = - \sum_{x^k \text{ mal classé}} y^k (a^T x^k + b)$$

avec $y^k = 1$ si $x^k \in C_1$ et $y^k = -1$ si $x^k \in C_2$.

Apprentissage

On utilise l'algorithme suivant :

1. on part de a_0 et b_0 aléatoires
2. on passe en revue les exemples :
   (a) si x^k est bien classé, on passe au suivant
   (b) sinon, on fabrique a_{t+1} et b_{t+1} grâce aux équations suivantes (avant de passer à l'exemple suivant) :

$$a_{t+1} = a_t + y^k x^k$$

$$b_{t+1} = b_t + y^k$$

On peut montrer que l'algorithme converge **si le problème est linéairement séparable**.

Discrimination linéaire : résumé

Réseau à une couche avec neurones avec fonction d'activation croissante :

- permet une discrimination linéaire
- les moindres carrés correspondent à la recherche d'une approximation de $P(C_j|x)$
- mais ils ne réalisent plus le maximum de vraisemblance
- les coefficients optimaux sont calculables dans le cas linéaire et dans certains cas particuliers non linéaires
- le modèle est optimal pour des données gaussiennes avec une matrice de covariance commune
- le modèle reste limité

Comme pour la régression linéaire, la discrimination linéaire constitue un bon modèle de référence.

Points généraux importants

Certains éléments abordés jusqu'à présent ne sont pas spécifiques aux réseaux à une couche :

- les moindres carrés correspondent à une estimation asymptotiquement correcte de $E(Y|x)$ ($P(C_j|x)$ en discrimination)
- pour la régression, les moindres carrés peuvent correspondre au maximum de vraisemblance (pas pour la classification)
- la discrimination optimale est obtenue en cherchant la classe la plus probable *a posteriori*
- la valeur de $\mathcal{E}(w)$ (ou le nombre d'erreurs de classement) ne suffit pas pour choisir le modèle optimal, sauf quand on a "beaucoup" de données