

Chapitre⁴ Méthode des moindres carrés

SAMIR KENOUCHE - DÉPARTEMENT DES SCIENCES DE LA MATIÈRE - UMKB

MÉTHODES NUMÉRIQUES ET PROGRAMMATION

Résumé

L'ajustement est une opération d'optimisation portant sur la recherche du profil théorique qui colle le mieux possible aux données expérimentales. Très souvent la représentation graphique s'avère insuffisante pour appréhender la relation fonctionnelle liant les données à ajuster. En effet, la connaissance a priori du modèle théorique peut s'avérer indispensable. Le processus d'ajustement doit permettre d'interpréter et de prévoir les variations de la variable dépendante (y_i) en fonction de la variable expliquée (x_i), supposée connue. De nombreuses méthodes existent pour ajuster les paramètres du modèle théorique afin de choisir le meilleur ajustement. Dans ce chapitre, nous aborderons l'ajustement au sens des moindres carrés ordinaires et pondérés. Ces techniques sont largement utilisées après la collecte de données que ce soit en Physique ou en Chimie. Dans ce chapitre il sera aussi question de l'étude de deux techniques de diagnostic de la régression.

Mots clés

Ajustement, moindres carrés, corrélation, valeur atypique, Matlab[®].

Table des matières

I Introduction	58
II Les moindres carrés ordinaires	59
II-A Intervalle de confiance	61
II-B Statistique de Student	62
III Les moindres carrés pondérés	68
IV Notions de corrélation	71
V Diagnostic de la régression	73
V-A Effet levier	73
V-B Distance de Cook	73
VI Travaux pratiques avec des fonctions Matlab prédéfinies	74

I. INTRODUCTION

L'objectif premier d'un ajustement des données expérimentales, est la recherche du modèle théorique en adéquation avec les données expérimentales. Dans certains cas, la représentation graphique est suffisante pour identifier la relation mathématique liant les données expérimentales à ajuster. Dans d'autres cas, la connaissance a priori du modèle théorique est inéluctable. L'ajustement doit permettre de décrire et de prévoir les variations de la variable dépendante (y_i) en fonction de la variable expliquée (x_i), supposée connue. De nombreuses méthodes existent pour ajuster les paramètres du modèle théorique afin de choisir le meilleur ajustement. Dans cette section, on abordera la

S. Kenouche est docteur en Physique de l'Université des Sciences et Techniques de Montpellier et docteur en Chimie de l'Université A. Mira de Béjaia.

Site web : voir <http://www.sites.univ-biskra.dz/kenouche>

Document corrigé, amélioré et actualisé le 19.09.2019.

régression linéaire¹ au sens **des moindres carrés**. D'un point de vue conceptuel, le modèle linéaire est caractérisé par deux entités déterministe et aléatoire selon :

$$\text{variable aléatoire} \leftarrow y = \underbrace{f(x; \theta_i)}_{\text{Déterministe}} + \underbrace{\epsilon_i}_{\text{Aléatoire}} \quad (1)$$

La variable dépendante y a un caractère aléatoire "hérité" de ϵ_i . Ce modèle de régression est construit en respectant les hypothèses suivantes :

- La distribution des résidus (ou erreur de régression) ϵ_i est indépendante de x . C'est l'hypothèse **d'indépendance**.
- La distribution des erreurs de régression suit une loi normale centré et de variance constante $\mathcal{N}(0, \sigma^2)$. Cette hypothèse porte aussi le nom d'**homoscédasticité** :

$$\forall i = 1, 2, \dots, n \quad \mathbb{E}(\epsilon_i) = 0, \quad \mathbb{V}(\epsilon_i) = \sigma^2$$

- $\text{Cov} [\epsilon_i, \epsilon_j] = 0$ pour $i \neq j$

Concrètement cette dernière hypothèse s'interprète en considérant :

$$y_1 = f(x; \theta_i) + \epsilon_1 \quad \text{et} \quad y_2 = f(x; \theta_i) + \epsilon_2 \quad (2)$$

Il n'existe aucune relation fonctionnelle entre les variables y_1 et y_2 . En d'autres mots, une variation de l'une des variables n'entraîne pas une variation prévisible pour l'autre.

II. LES MOINDRES CARRÉS ORDINAIRES

Le fondement de la méthode des moindres carrés est régit par la minimisation de la somme quadratique des écarts (encore appelés résidus) entre les données expérimentales et le modèle considéré. Le but consiste à trouver le modèle théorique, de forme générale $f(x_i; \theta_i)$, qui ajuste le mieux les mesures expérimentales. Avec, θ_i sont les paramètres du modèle en question, ayant une signification physique. Dans la méthode des moindres carrés, ces derniers sont déterminés par minimisation d'une *fonction-objectif* (appelée aussi *fonction de coût* ou *critère d'optimisation*), notée $S(\hat{\theta}_1, \hat{\theta}_2)^2$. Ainsi, le critère d'optimalité est celui de la minimisation des résidus. Les estimations, $\hat{\theta}_1$ et $\hat{\theta}_2$, traduisent le minimum de la *fonction-objectif*. Nous allons illustrer ce propos, en considérant dans un premier temps le modèle : $\hat{y}_i = \theta_1 x_i + \theta_2$

$$S(\hat{\theta}_1, \hat{\theta}_2) = \sum_i^n e_i^2 = \sum_i^n (y_i - f(x_i; \theta_i))^2 \quad (3)$$

$$S(\hat{\theta}_1, \hat{\theta}_2) = \sum_i^n e_i^2 = \sum_i^n (y_i - \theta_1 \times x_i - \theta_2)^2 \quad (4)$$

1. Le mot linéaire ici s'applique aux coefficients et non pas à la variable explicative. Dans ce cas $\frac{\partial f(a_i, x_i)}{\partial a_i} \neq f(a_i)$. Pour un modèle non-linéaire on aura $\frac{\partial f(a_i, x_i)}{\partial a_i} = f(a_i)$.

2. Il convient de remarquer que la fonction $S(\hat{\theta}_1, \hat{\theta}_2)$ (voir l'équation (4)) est strictement convexe. Elle admet un minimum en un unique point $(\hat{\theta}_1$ et $\hat{\theta}_2)$, lequel est obtenu en annulant les dérivées partielles de S .

Cette expression traduit la somme de toutes les distances verticales, dont l'unité est celle des abscisses, entre les données expérimentales et le modèle théorique considéré. Mathématiquement, cette minimisation s'exprime par :

$$\begin{cases} \left. \frac{\delta S(\theta_1, \theta_2)}{\delta \theta_1} \right|_{\theta_1 = \hat{\theta}_1} = \sum_{i=1}^n 2(y_i - \hat{\theta}_1 \times x_i - \hat{\theta}_2) \times (-x_i) = 0 \\ \left. \frac{\delta S(\theta_1, \theta_2)}{\delta \theta_2} \right|_{\theta_2 = \hat{\theta}_2} = \sum_{i=1}^n 2(y_i - \hat{\theta}_1 \times x_i - \hat{\theta}_2) \times (-1) = 0 \end{cases} \quad (5)$$

$$\Rightarrow \begin{cases} 2 \left[\sum_{i=1}^n y_i - \hat{\theta}_1 \sum_{i=1}^n x_i - \hat{\theta}_2 \right] \times (-x_i) = 0 \\ 2 \left[\sum_{i=1}^n y_i - \hat{\theta}_1 \sum_{i=1}^n x_i - \hat{\theta}_2 \right] \times (-1) = 0 \end{cases} \quad (6)$$

$$\Rightarrow \begin{cases} 2 \left[-\sum_{i=1}^n y_i x_i + \hat{\theta}_1 \sum_{i=1}^n x_i^2 + \hat{\theta}_2 \sum_{i=1}^n x_i \right] = 0 \\ 2 \left[-\sum_{i=1}^n y_i + \hat{\theta}_1 \sum_{i=1}^n x_i + \hat{\theta}_2 \right] = 0 \end{cases} \quad (7)$$

$$\Rightarrow \begin{cases} \hat{\theta}_1 \sum_{i=1}^n x_i^2 + \hat{\theta}_2 \sum_{i=1}^n x_i = \sum_{i=1}^n y_i x_i \\ \hat{\theta}_1 \sum_{i=1}^n x_i + \hat{\theta}_2 = \sum_{i=1}^n y_i \end{cases} \quad (8)$$

Ce système d'équations peut s'écrire sous forme matricielle, selon :

$$\begin{bmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & 1 \end{bmatrix} \times \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n y_i \end{bmatrix} \quad (9)$$

Le vecteur des paramètres estimés s'obtient :

$$\begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & 1 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n y_i \end{bmatrix} \quad (10)$$

Rappelons que l'inverse d'une matrice 2×2 se calcule simplement par la formule générale :

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \Rightarrow M^{-1} = \frac{1}{\det(M)} \begin{bmatrix} D & -B \\ -C & A \end{bmatrix} \quad (11)$$

Avec $\det(M) = AD - BC$

La pente $\hat{\theta}_1$ et l'ordonnée à l'origine $\hat{\theta}_2$ sont déterminées à partir des formules :

$$\hat{\theta}_1 = \frac{\sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i\right)^2} \quad \text{et} \quad \hat{\theta}_2 = \frac{\sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i - \sum_{i=1}^n x_i \sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i\right)^2}$$

A. Intervalle de confiance

On donne sans démonstration les incertitudes commises sur les deux paramètres d'ajustement en question :

$$\Delta\hat{\theta}_1 = t_{n-2,\alpha} \times s_{\hat{\theta}_1} = t_{n-2,\alpha} \times \frac{s_r}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}} \tag{12}$$

$$\Delta\hat{\theta}_2 = t_{n-2,\alpha} \times s_{\hat{\theta}_2} = t_{n-2,\alpha} \times s_r \times \sqrt{\frac{\sum_{i=1}^n x_i^2}{n \sum_{i=1}^n (x_i - \bar{x})^2}} \tag{13}$$

Avec $t_{n-2,\alpha}$ (voir la table (VI)) est le quantile de la loi de Student à $n - 2$ degrés de liberté, pour un niveau de confiance $1 - \alpha$ et s_r est l'écart-type sur les résidus, donnée par :

$$s_r = \sqrt{\frac{1}{n-2} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \tag{14}$$

On l'appelle aussi l'erreur quadratique moyenne. Plus la valeur de s_r est faible, meilleur est l'ajustement. A partir des equations ci-dessus, on comprend sans peine que plus l'écart-type des coefficients est faible, plus les valeurs estimées de $\hat{\theta}_1$ et $\hat{\theta}_2$ sont précises. Ceci appelle les observations suivantes :

- Les mesures x_i doivent être dispersées autour de leur moyenne \Rightarrow la quantité statistique $\sum_{i=1}^n (x_i - \bar{x})^2$ doit être la plus grande possible.
- Le terme $\sum_{i=1}^n x_i^2$ doit être le plus faible possible. Les x_i doivent présenter une faible moyenne en valeur absolue.
- L'erreur quadratique moyenne s_r doit être la plus faible possible. Plus s_r est faible, plus la droite de régression est proche des points expérimentaux.
- Plus il y a de points expérimentaux (n grand), meilleure est la précision des paramètres d'ajustement.
- L'incertitude affectant les paramètres d'ajustement dépend aussi de niveau de confiance fixé ($1 - \alpha$) à travers le coefficient de Student $t_{n-2,\alpha}$.

La méthode des moindres carrés fournit des variances minimales pour les coefficients estimés. Cette propriété est stipulé par le théorème de Gauss-Markov. La démonstration de ce théorème dépasse largement le cadre de ce cours. Le plus important pour un(e) étudiant(e) en sciences expérimentales, est d'avoir une démarche compréhensible et rigoureuse afin d'analyser ce type de problématique.

B. Statistique de Student

Le test de signification des coefficients est réalisé au moyen du test de Student³. Un coefficient est dit significatif si la variable qui lui est associée a une influence sur la réponse. Dans ce cas de figure, la p-value est la probabilité qu'un coefficient soit négligeable. On calcule cette p-value à partir du rapport du coefficient à son écart-type :

$$t_{obs} = \frac{|\hat{\theta}_i|}{s_{\hat{\theta}_i}} \tag{15}$$

On testera à ce propos, les deux hypothèses suivantes :

$$\begin{cases} \mathcal{H}_0 \implies \hat{\theta}_i = 0 & \text{coefficient non significatif} \\ \mathcal{H}_1 \implies \hat{\theta}_i \neq 0 & \text{coefficient significatif} \end{cases} \tag{16}$$

Si $p\text{-value} = \mathbb{P}_{\mathcal{H}_0}(T > t_{obs}) < \alpha \implies$ l'hypothèse \mathcal{H}_0 est rejetée au risque 5%. Dans les logiciels de statistique, la probabilité $\mathbb{P}_{\mathcal{H}_0}(T > t_{obs})$ est calculée selon :

$$p\text{-value} = \mathbb{P}(T > t_{obs}) = 1 - \mathbb{P}(T < t_{obs}) = 1 - \Phi(t_{obs}) \tag{17}$$

Avec $\Phi(t_{obs})$ est la fonction de répartition de la loi de Student ou encore la distribution cumulative de la loi de Student (Student's t cumulative distribution function). L'hypothèse H_0 est acceptée dans le cas où :

$$1 - \Phi(t_{obs}) \geq 0.05 \tag{18}$$

et elle est rejetée si

$$1 - \Phi(t_{obs}) < 0.05 \tag{19}$$

Dans le cas où l'hypothèse \mathcal{H}_0 est acceptée, cela veut dire que le coefficient en question n'est pas, au risque α , significativement différent de zéro. Ainsi, la variable qui lui est associée n'a pas d'influence sur la réponse. L'allure de la distribution de la loi de Student et de celle de sa fonction de répartition sont représentées ci-dessous.

Pour les étudiants (es) qui ont du mal à se familiariser avec le calcul des probabilités, peuvent comparer directement le quantile $t_{n-2,\alpha}$ (voir la table (VI)) à celui calculé à partir de la statistique de Student.

Le test impose : $\mathbb{P}(T > t_{n-2,\alpha}) = \alpha$

Si $t_{obs} > t_{n-2,\alpha} \implies \mathbb{P}(T > t_{obs}) < \mathbb{P}(T > t_{n-2,\alpha}) \implies \mathbb{P}(T > t_{obs}) < \alpha$

L'hypothèse \mathcal{H}_0 est rejetée au risque α

Si $t_{obs} < t_{n-2,\alpha} \implies \mathbb{P}(T > t_{obs}) > \mathbb{P}(T > t_{n-2,\alpha}) \implies \mathbb{P}(T > t_{obs}) > \alpha$

L'hypothèse \mathcal{H}_0 est acceptée au risque α

3. On l'appelle aussi test de nullité d'un coefficient.

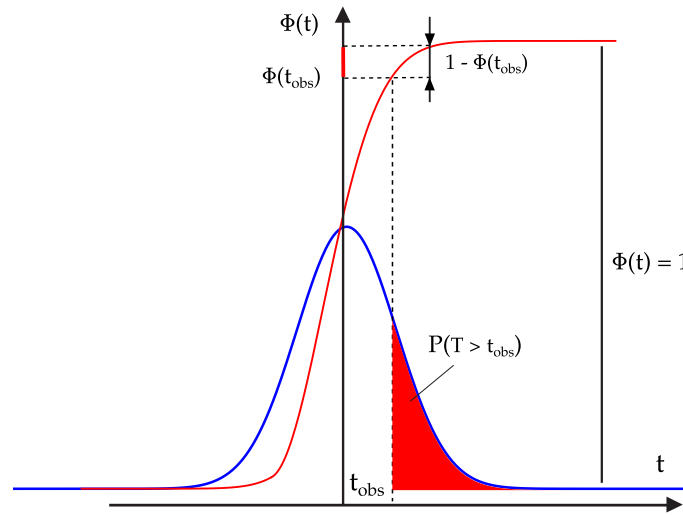


FIGURE 1: Distribution et fonction de répartition de la loi de Student

Exercice

Nous avons mesuré la conductivité ($mSiemens/cm^2$) d'un électrolyte en fonction de sa concentration. Les mesures ont été réalisées à $T = 25 C^\circ$. On rappelle que la conductivité d'une solution s'écrit sous la forme :

$$\sigma = \sum_{i=1}^n c_i \lambda_i \tag{20}$$

Avec, n étant le nombre d'ions (cation et anion) présents dans l'électrolyte, c_i est la concentration de l'ion i de l'électrolyte et λ_i est sa conductivité molaire ionique. Les résultats obtenus sont :

$$\sigma = [1.04 , 1.98 , 3.38 , 2.94 , 4.72 , 5.12 , 5.84 , 6.05 , 7.80 , 9.29 , 9.40 , 9.37 , 11.03 , 14.1 , 13.7 , 13.4 , 14.65]$$

La gamme des concentrations vaut [1 : 1 : 17] (mmol/L).

- Déterminer la droite de régression du nuage de points (c_i, σ_i) .
- Déterminer l'intervalle de confiance des paramètres d'ajustement.
- Réaliser le Test de nullité d'un coefficient (ou test de Student). Commenter
- Calculer le coefficient de corrélation du nuage de points.
- Commenter le résultat obtenu.

Solution : commençons par appliquer les relations calculant $\hat{\theta}_1$ et $\hat{\theta}_2$

$$\hat{\theta}_1 = \frac{17 \times 1.55 \cdot 10^3 - 153 \times 133.81}{30345 - 23409} \quad \text{et} \quad \hat{\theta}_2 = \frac{1785 \times 133.81 - 153 \times 1.55 \cdot 10^3}{30345 - 23409}$$

$$\Rightarrow \hat{\theta}_1 = 0.8698 \quad \text{et} \quad \hat{\theta}_2 = 0.0427$$

$$s_r = 0.72 \quad ; \quad s_{\hat{a}} = 0.04 \quad ; \quad s_{\hat{b}} = 0.36 \quad ; \quad t_{15} = 1.75 \text{ à } 95\%$$

TABLE I: Calcul statistique

x_i	x_i^2	y_i	$x_i y_i$
1.0000	1.0000	1.0400	1.0400
2.0000	4.0000	1.9800	3.9600
3.0000	9.0000	3.3800	10.1400
4.0000	16.0000	2.9400	11.7600
5.0000	25.0000	4.7200	23.6000
6.0000	36.0000	5.1200	30.7200
7.0000	49.0000	5.8400	40.8800
8.0000	64.0000	6.0500	48.4000
9.0000	81.0000	7.8000	70.2000
10.0000	100.0000	9.2900	92.9000
11.0000	121.0000	9.4000	103.4000
12.0000	144.0000	9.3700	112.4400
13.0000	169.0000	11.0300	143.3900
14.0000	196.0000	14.1000	197.4000
15.0000	225.0000	13.7000	205.5000
16.0000	256.0000	13.4000	214.4000
17.0000	289.0000	14.6500	249.0500
$\sum_{i=1}^{17} x_i = 153$	$\sum_{i=1}^{17} x_i^2 = 1785$	$\sum_{i=1}^{17} y_i = 133.81$	$\sum_{i=1}^{17} x_i y_i = 1.55 \cdot 10^3$

TABLE II: Résultat de la régression

y_i	\hat{y}_i	$e_i = y_i - \hat{y}_i$	e_i^2
1.0400	0.9030	0.1370	0.0188
1.9800	1.7630	0.2170	0.0471
3.3800	2.6230	0.7570	0.5730
2.9400	3.4830	-0.5430	0.2948
4.7200	4.3430	0.3770	0.1421
5.1200	5.2030	-0.0830	0.0069
5.8400	6.0630	-0.2230	0.0497
6.0500	6.9230	-0.8730	0.7621
7.8000	7.7830	0.0170	0.0003
9.2900	8.6430	0.6470	0.4186
9.4000	9.5030	-0.1030	0.0106
9.3700	10.3630	-0.9930	0.9860
11.0300	11.2230	-0.1930	0.0372
14.1000	12.0830	2.0170	4.0683
13.7000	12.9430	0.7570	0.5730
13.4000	13.8030	-0.4030	0.1624
14.6500	14.6630	-0.0130	0.0002

$$\Rightarrow \Delta \hat{\theta}_1 = 2.13 \times 0.04 = 0.06$$

$$\Rightarrow \Delta \hat{\theta}_2 = 2.13 \times 0.36 = 0.64$$

La droite de régression s'écrit comme suit :

$$\hat{y} = 0.87 (\pm 0.06) \times x + 0.04 (\pm 0.64)$$

Les intervalles de confiance s'écrivent également sous la forme :

$$\hat{\theta}_1 = 0.87 [0.81 \text{ à } 0.93] \quad \text{et} \quad \hat{\theta}_2 = 0.04 [-0.60 \text{ à } 0.68]$$

Qui se lit comme suit, nous avons 95 chances sur 100 de trouver la vraie valeur (θ_1) de $\hat{\theta}_1$ dans l'intervalle [0.81 à 93]. Le paramètre estimé $\hat{\theta}_1 = 0.87$ étant la valeur la plus probable. Il en est de même pour l'ordonnée à l'origine. Si l'on souhaite affiner encore cette analyse, on testera la signification de chaque paramètre d'ajustement.

A l'issue de cette étape le coefficient n'ayant pas d'influence sur la réponse (la conductivité) sera éliminé de l'équation finale. D'ores et déjà rien qu'en comparant l'erreur commise sur le paramètre avec le paramètre lui-même, on peut tirer des conclusions sur son influence ou non sur la réponse. Calculons le test de Student pour $\hat{\theta}_1$ et $\hat{\theta}_2$:

$$t_1 = \frac{|\hat{\theta}_1|}{s_{\hat{\theta}_1}} = \frac{0.87}{0.04} = 21.75 \quad \text{et} \quad t_2 = \frac{|\hat{\theta}_2|}{s_{\hat{\theta}_2}} = \frac{0.04}{0.63} = 0.06$$

```
clc; clear all
% Samir Kenouche le 06/05/2019
% Test de Student, par défaut alpha = 0.05

n = 17; t1 = 0.87/0.04; t2 = 0.04/0.63;
t = [t1 , t2];

for it = 1:numel(t)
pvalue(it) = 1 - tcdf(t(it), n-1)
end
```

On trouve une p-value = $1.30 \cdot 10^{-13}$ pour $\hat{\theta}_1$ et p-value = 0.47 pour $\hat{\theta}_2$. La question qui se pose désormais est : comment interpréter les résultats de la p-value ? En d'autres mots, quels sont les coefficients statistiquement significatifs ? A partir des résultats de la p-value, on trouve pour le coefficient $\hat{\theta}_1$, une p-value valant $1.30 \cdot 10^{-13}$. Cela signifie qu'elle a moins de 1 chance sur $\simeq 10^{13}$ d'avoir une valeur égale à zéro. Le coefficient $\hat{\theta}_1$ est donc significativement différent de zéro. Il faut donc le conserver dans le modèle prédictif. Le coefficient $\hat{\theta}_2$ a une p-value = 0.47. Cela signifie qu'il a 47 chances sur 100 d'avoir une valeur égale à zéro. En effet, ce coefficient a donc une forte chance d'être égale à zéro par rapport au seuil fixé $\alpha = 0.05$. Il n'est donc pas significativement différent de zéro (ou très peu) et on peut l'éliminer du modèle prédictif. Il est possible aussi d'interpréter ce résultat en disant que la présence ou l'absence de ce coefficient dans le modèle prédictif n'a pas d'effet (ou très peu) sur les réponses calculées. Ainsi, tenant compte des valeurs des p-value calculées, le modèle prédictif s'écrit :

$$y = 0.87 (\pm 0.06) x \quad \text{à} \quad 95\% \tag{21}$$

Il est possible désormais de représenter graphiquement cette droite ($y = 0.87 x$) avec son intervalle de confiance suivant les deux équations suivantes $y_{min} = 0.81 x$ et $y_{max} = 0.93 x$.

L'intervalle de confiance représenté sur le graphe s'interprète comme suit. Si nous mesurons à nouveau un couple (c_i, σ_i) , nous avons 95% de chance pour que ce point expérimental soit à l'intérieur de l'intervalle, délimité par les droites extrêmes y_{min} et y_{max} . Si nous répétons n fois cette mesure, 5% des couples (c_i, σ_i) seront en dehors de cet intervalle. Par ailleurs, la droite en rouge ($y = 0.87 x$) étant la plus probable.

En guise de conclusion, à partir de cette étude expérimentale, nous avons collecté n couples (x_i, y_i) . Ensuite nous avons modélisé ces points expérimentaux par une droite affine. A l'issue de cette modélisation (ou ajustement), les paramètres $\hat{\theta}_1$ et $\hat{\theta}_2$ ont été estimés par la méthode des **moindres carrés ordinaire**. Si maintenant nous répétons la même expérience, nous mesurerions n nouveaux couples de données (x_i, y_i) . A partir de ces données, nous déterminerions de nouveaux paramètres $\hat{\theta}_1$ et $\hat{\theta}_2$. Ainsi, ces derniers changent en fonction des données recueillies. Les vraies valeurs de ces paramètres d'ajustement θ_1 et θ_2 ne changent pas et sont inaccessibles par l'expérience.

En analyse de données on essaye systématiquement de diminuer le biais (l'écart) entre les vrais paramètres (θ_1, θ_2) et ceux estimés $(\hat{\theta}_1, \hat{\theta}_2)$ par une méthode statistique. Cela se traduit par l'écriture $\mathbb{E}(\hat{\theta}_1) = \theta_1$ et $\mathbb{E}(\hat{\theta}_2) = \theta_2$.

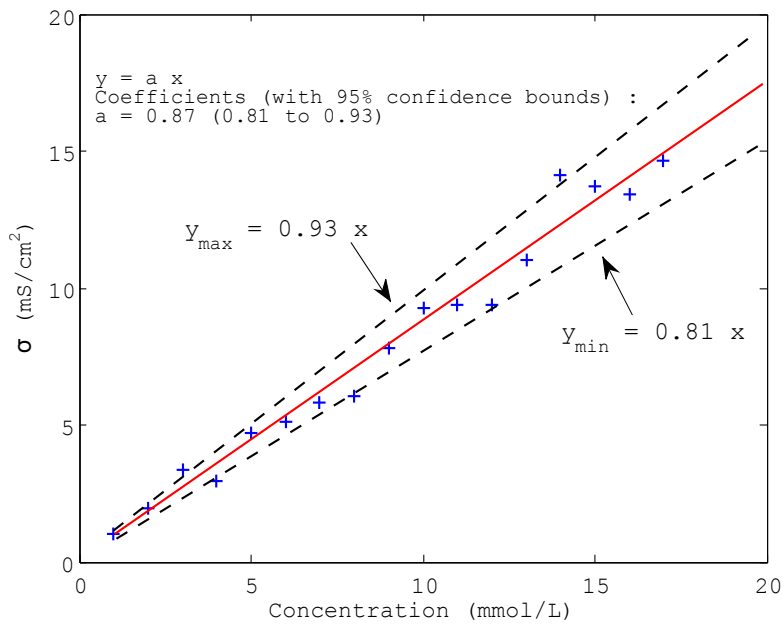


FIGURE 2: Droite de régression (en rouge) et intervalle de confiance (en pointillés)

Ci-dessous, l'implémentation au moyen d'un calcul analytique de la méthode des moindres carrés ordinaires.

```

clc ; clear all ; close all ;
% Samir KENOUCHE - le 06/09/2019
% METHODE DES MOINDRES CARRES ORDINAIRES : CALCUL ANALYTIQUE

x = 1:17 ;

y = [1.04 , 1.98 , 3.38 , 2.94 , 4.72 , 5.12 , 5.84 , 6.05 , 7.80 , ...
     9.29 , 9.40 , 9.37 , 11.03 , 14.10 , 13.70 , 13.40 , 14.65] ;

n = numel(y) ; % TAILLE DES DONNEES
num_theta1 = n*sum(x.*y) - sum(x)*sum(y) ; % NUMERATEUR - theta1
num_theta2 = sum(x.^2)*sum(y) - sum(x)*sum(x.*y) ; % NUMERATEUR - theta2
den = n.*sum(x.^2) - (sum(x)).^2 ; % DENOMINATEUR

theta1 = num_theta1/den ; % LA PENTE
theta2 = num_theta2/den ; % ORDONNEE A L'ORIGINE

yhat = theta1.*x + theta2 ; % CONSTRUCTION DU MODELE
sr = sqrt(sum((y - yhat).^2)*(n-2)) ; % ERREUR DE REGRESSION

s_theta1 = sr/sqrt(sum((x - mean(x)).^2)) ; % ECART-TYPE POUR LA PENTE
s_theta2 = sr*sqrt(sum(x.^2)/(n*sum((x - mean(x)).^2))) ; % ECART-TYPE POUR
ORDONNEE A L'ORIGINE

p = 0.95 ; t = tinv(p, n-2) ; % QUANTILE DE LA LOI DE STUDENT

delta_theta1 = t*s_theta1 ; % INCERTITUDE SUR LA PENTE
delta_theta2 = t*s_theta2 ; % INCERTITUDE SUR L'ORDONNEE A L'ORIGINE
    
```

Ci-dessous, l'implémentation de la méthode des moindres carrés ordinaires avec des fonctions Matlab® prédéfinies.

```

clc ; clear all ; close all ;
% Samir KENOUCHE - le 06/09/2019
% METHODE DES MOINDRES CARRES ORDINAIRES
% AVEC LES FONCTIONS : polyfit et polyval

xData = 1:17 ;

yData = [1.04 , 1.98 , 3.38 , 2.94 , 4.72 , 5.12 , 5.84 , 6.05 , 7.80, ...
         9.29 , 9.40 , 9.37 , 11.03 , 14.10 , 13.70 , 13.40 , 14.65] ;

n = 1 ; % POLYNOME DE DEGRE UN
coeffs = polyfit(xData, yData, n) ; % RECUPERATION DES COEFFICIENTS
modelFun = polyval(coeffs, xData) ; % EVALUATION DU POLYNOME
plot(xData, yData, 'o') ; hold on ; plot(xData, modelFun, 'r') ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AFFICHAGE SUR LA FIGURE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
h = gca ;
str(1) = {'Plot of the model :'} ;
str(2) = {'$$y = \hat{\theta}_1 x + \hat{\theta}_2$$'} ;
str(3) = {'With the values:'} ;
str(3) = [{'$$\hat{\theta}_1 = $$', num2str(coeffs(2))}] ;
str(4) = [{'$$\hat{\theta}_2 = $$', num2str(coeffs(1))}] ;
set(gcf, 'CurrentAxes', h)
text('Interpreter', 'latex', 'String', str, 'Position', [0.50 12], ...
'FontSize', 12)
    
```

```

clc ; clear all ; close all ;
% Samir KENOUCHE - le 06/09/2019
% METHODE DES MOINDRES CARRES ORDINAIRES
% AVEC LA FONCTION : regress

xData = 1:17 ;

yData = [1.04 , 1.98 , 3.38 , 2.94 , 4.72 , 5.12 , 5.84 , 6.05 , 7.80, ...
         9.29 , 9.40 , 9.37 , 11.03 , 14.10 , 13.70 , 13.40 , 14.65] ;

xn = [xData ; ones(size(xData))]' ;
[coeffs, Bint, residual] = regress(yData', xn) ;
modelFun = coeffs(1).*xData + coeffs(2) ;

plot(xData, yData, 'o') ; hold on ; plot(xData, modelFun, 'r')

% bint : returns a matrix of 95% confidence intervals for the
% coefficient estimates. The first column of bint contains lower
% confidence bounds for each of the p coefficient estimates; the
% second column contains upper confidence bounds
    
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% INTERVALLE DE CONFIANCE - theta1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A1 = Bint(1,1); A2 = 'To' ; A3 = Bint(1,2); formatSpec = '%3$s %2$s %1$s';
pente = sprintf(formatSpec, A3, A2, A1)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% INTERVALLE DE CONFIANCE - theta2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
B1 = Bint(1,2); B2 = 'To' ; B3 = Bint(2,2); formatSpec = '%3$s %2$s %1$s';
ordonnee = sprintf(formatSpec, B1, B2, B3)
    
```

```

clc ; clear all ; close all ;
% Samir KENOUCHE - le 06/09/2019
% METHODE DES MOINDRES CARRÉS ORDINAIRES
% AVEC LA FONCTION : regstats

xData = 1:17 ;

yData = [1.04 , 1.98 , 3.38 , 2.94 , 4.72 , 5.12 , 5.84 , 6.05 , 7.80, ...
        9.29 , 9.40 , 9.37 , 11.03 , 14.10 , 13.70 , 13.40 , 14.65] ;

% xn = [xData ; ones(size(xData))]' ;
% hii = diag(xn*inv(xn'*xn)*xn') ;
n = numel(yData) ; mystats = regstats(yData, xData,'linear') ;

theta_1 = mystats.beta(1) ;
theta_2 = mystats.beta(2) ;
Rsquare = mystats.rsquare ;
levier = mystats.leverage ; % EFFET LEVIER
cook_d = mystats.cookd ; % DISTANCE DE COOK

s_theta1 = sqrt(mystats.covb(1,1)) ; % ECART-TYPE SUR LA PENTE
s_theta2 = sqrt(mystats.covb(2,2)) ; % ECART-TYPE SUR L'ORDONNEE

p = 0.95 ; t = tinv(p, n-2) ; % QUANTILE DE LA LOI DE STUDENT

delta_theta1 = t*s_theta1 ; % INCERTITUDE SUR LA PENTE
delta_theta2 = t*s_theta2 ; % INCERTITUDE SUR L'ORDONNEE
    
```

III. LES MOINDRES CARRÉS PONDÉRÉS

L'estimateur des moindres carrés est modifié, et la somme des carrés des écarts est pondérée par le poids de chaque point expérimental :

$$S(\hat{\theta}_1, \hat{\theta}_2) = \sum_i^n w_i e_i^2 = \sum_i^n w_i (y_i - \theta_1 \times x_i - \theta_2)^2 \tag{22}$$

On s'intéressera au cas où la variable dépendante est entachée d'une incertitude $y_i \pm \Delta y_i$ et la variable explicative x_i est déterministe $\Delta x_i \rightarrow 0$. En outre, on abordera le cas où l'incertitude Δy_i n'est pas la même pour tout les y_i .

4. L'incertitude correspond à l'écart-type des y_i .

On donne sans démonstration les relations suivantes :

$$\hat{\theta}_1 = \frac{\sum_{i=1}^n w_i \sum_{i=1}^n w_i x_i y_i - \sum_{i=1}^n w_i x_i \sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i \sum_{i=1}^n w_i x_i^2 - \left(\sum_{i=1}^n w_i x_i\right)^2} \quad (23)$$

$$\hat{\theta}_2 = \frac{\sum_{i=1}^n w_i y_i \sum_{i=1}^n w_i x_i^2 - \sum_{i=1}^n w_i x_i \sum_{i=1}^n w_i x_i y_i}{\sum_{i=1}^n w_i \sum_{i=1}^n w_i x_i^2 - \left(\sum_{i=1}^n w_i x_i\right)^2} \quad (24)$$

Les σ_{y_i} étant connus, en appliquant la formule de propagation des incertitudes⁵, il vient :

$$\Delta \hat{\theta}_1 = \sqrt{\frac{\sum_{i=1}^n w_i}{\sum_{i=1}^n w_i \sum_{i=1}^n w_i x_i^2 - \left(\sum_{i=1}^n w_i x_i\right)^2}} \quad (25)$$

$$\Delta \hat{\theta}_2 = \sqrt{\frac{\sum_{i=1}^n w_i x_i^2}{\sum_{i=1}^n w_i \sum_{i=1}^n w_i x_i^2 - \left(\sum_{i=1}^n w_i x_i\right)^2}} \quad (26)$$

$$w_i = \frac{1}{\sigma_{y_i}^2} \quad (27)$$

Chaque point expérimental est pondéré par son poids à travers la quantité w_i . Plus la pondération w_i est grande (donc σ_{y_i} est petit) meilleure est la précision sur les paramètres d'ajustement. La droite de régression passera en "priorité" par les points expérimentaux ayant la plus faible incertitude. Autrement dit, les points de faible incertitude auront plus de poids dans l'établissement de la droite de régression.

$$\sum_{i=1}^{17} x_i = 153.0 \quad \text{et} \quad \sum_{i=1}^{17} y_i = 133.8$$

$$\hat{\theta}_1 = \frac{2.23 \cdot 10^4 - 2.09 \cdot 10^4}{4.86 \cdot 10^4 - 2.35 \cdot 10^4} \Rightarrow \hat{\theta}_1 = 0.85 \quad \text{et} \quad \Delta \hat{\theta}_1 = \sqrt{\frac{33.35}{2.50 \cdot 10^4}} \Rightarrow \Delta \hat{\theta}_1 = 0.04$$

$$\hat{\theta}_2 = \frac{1.99 \cdot 10^5 - 1.94 \cdot 10^5}{4.86 \cdot 10^4 - 2.35 \cdot 10^4} \Rightarrow \hat{\theta}_2 = 0.16 \quad \text{et} \quad \Delta \hat{\theta}_2 = \sqrt{\frac{1.45 \cdot 10^3}{2.50 \cdot 10^4}} \Rightarrow \Delta \hat{\theta}_2 = 0.24$$

$$t_1 = \frac{|\hat{\theta}_1|}{\sigma_{\hat{\theta}_1}} = \frac{0.85}{0.04} = 21.25 \quad \text{et} \quad t_2 = \frac{|\hat{\theta}_2|}{\sigma_{\hat{\theta}_2}} = \frac{0.16}{0.24} = 0.66$$

$t_1 > t_{(15,0.05)}$ ⁶ $\Rightarrow \mathcal{H}_0$ est rejetée au risque 5% \Rightarrow le coefficient $\hat{\theta}_1$ est conservé

5. Pour plus de détails sur cette formule, voir mon cours intitulé "Méthodes d'analyse quantitative"

6. Quantile de la loi de Student $t_{15,0.05} = 1.75$, voir la table (VI).

TABLE III: Calcul statistique

σ_{y_i}	w_i	$w_i x_i$	$w_i y_i$	$w_i x_i y_i$
0.3000	11.1111	11.1111	11.5556	11.5556
0.4000	6.2500	12.5000	12.3750	24.7500
0.7000	2.0408	6.1224	6.8980	20.6939
0.5000	4.0000	16.0000	11.7600	47.0400
0.9000	1.2346	6.1728	5.8272	29.1358
1.0500	0.9070	5.4422	4.6440	27.8639
1.1500	0.7561	5.2930	4.4159	30.9112
1.2300	0.6610	5.2879	3.9989	31.9915
1.2800	0.6104	5.4932	4.7607	42.8467
0.9000	1.2346	12.3457	11.4691	114.6914
1.7000	0.3460	3.8062	3.2526	35.7785
1.6000	0.3906	4.6875	3.6602	43.9219
1.7500	0.3265	4.2449	3.6016	46.8212
1.4500	0.4756	6.6587	6.7063	93.8882
1.2700	0.6200	9.3000	8.4940	127.4103
0.8000	1.5625	25.0000	20.9375	335.0000
1.1000	0.8264	14.0496	12.1074	205.8264
/	$\sum_{i=1}^{17} w_i = 33.4$	$\sum_{i=1}^{17} w_i x_i = 153.5$	$\sum_{i=1}^{17} w_i y_i = 136.5$	$\sum_{i=1}^{17} w_i x_i y_i = 1270$

$t_2 < t_{(15,0.05)} \Rightarrow \mathcal{H}_0$ est acceptée au risque 5% \Rightarrow le coefficient $\hat{\theta}_2$ est éliminé

Tenant compte du test de Student, le modèle de prédiction final s'écrit :

$$\hat{y} = 0.85 (\pm 0.04) x \tag{28}$$

Ci-dessous, la droite de régression déterminée à partir de la méthode des moindres carrés pondérés (MCP). En comparant cette droite à celle obtenue par la méthode des MCO (figure (2)), on constate un rétrécissement de l'intervalle de confiance de la pente. La largeur de cet intervalle passe de 0.26 à 0.16. Ainsi, le coefficient $\hat{\theta}_1$ gagne en précision (environ deux fois plus) avec la méthode des MCP.

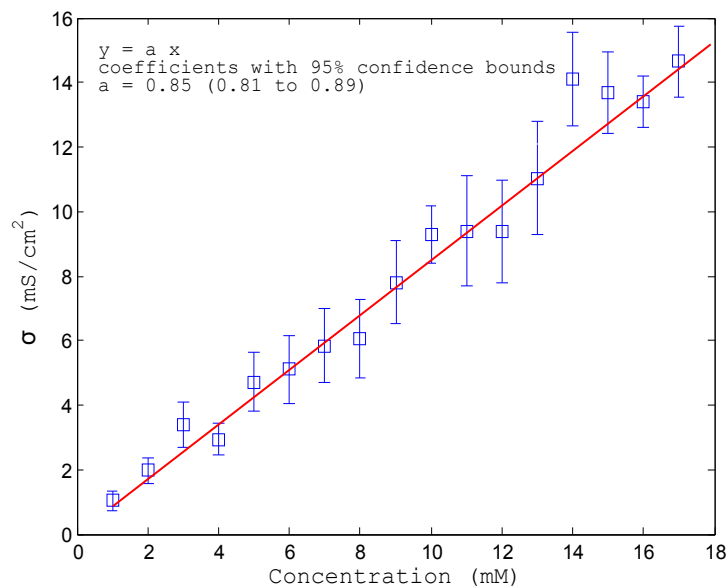


FIGURE 3: Droite de régression obtenue par la méthode des MCP

IV. NOTIONS DE CORRÉLATION

La corrélation du nuage de point (x_i, y_i) est caractérisée par un coefficient numérique appelé **coefficient de corrélation linéaire**. On l'appelle aussi le coefficient de corrélation de Pearson. Ce dernier est indépendant des unités de mesure. Empiriquement ce coefficient s'obtient :

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (29)$$

Avec une écriture plus compacte, on aura :

$$\text{Corr}(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} \quad (30)$$

Le coefficient de corrélation r est toujours compris entre -1 et $+1$

$$-1 \leq r \leq +1 \quad (31)$$

Les figures ci-dessous présentent diverses configurations du nuage de points (Scatter Plot du nuage (x_i, y_i)) avec le coefficient de corrélation correspondant.

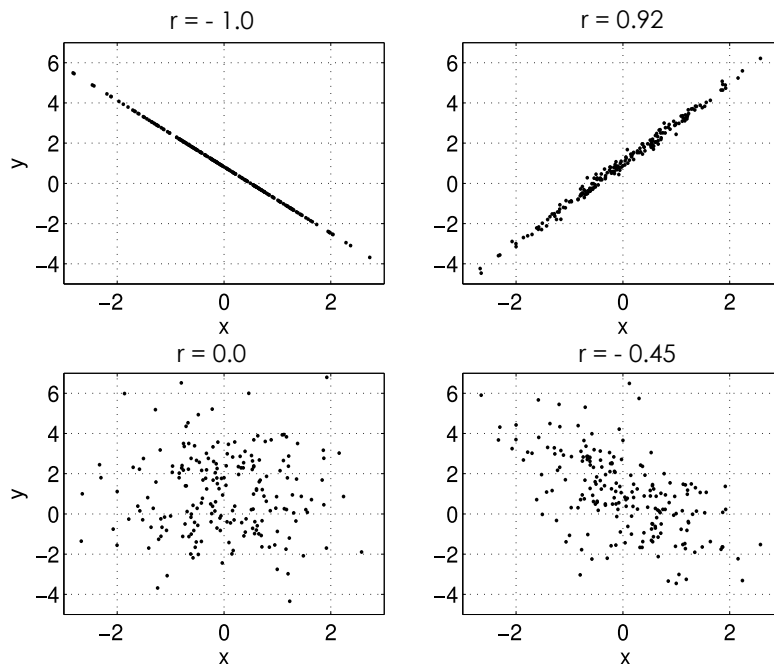


FIGURE 4: Quelques liaisons (x_i, y_i)

- Si $r = +1$ (ou -1), cela veut dire que la dépendance entre X et Y est totale. Il existe une relation linéaire parfaite entre les deux grandeurs.
- Si $r = 0$, il n'existe aucune corrélation linéaire entre les deux grandeurs. En revanche, il peut y avoir une corrélation non linéaire.

- Si $r > 0$, cela implique que X et Y évoluent dans le même sens. Inversement, si $r < 0$, les deux grandeurs varient en sens inverse.

Il est d'usage de considérer un $|r| > 0.9$ comme une liaison linéaire forte. Il est déconseillé en revanche d'ajuster une droite à un nuage de points quand le coefficient de corrélation linéaire vaut $|r| < 0.65$.

Important quand le nuage de points ne s'effiloche pas autour d'une droite mais qu'une tendance semble monotone (croissance ou décroissance), on utilise dans ce cas le coefficient de **Spearman**. Il a été constaté également que le coefficient de Spearman résiste mieux aux valeurs atypiques contrairement au coefficient de Pearson. Pour des données ordinales, on utilise le coefficient de **Kendall**.

Par ailleurs, certains modèles non-linéaires sont linéarisables. Ci-dessous quelques cas usuels de linéarisation :

TABLE IV: Transformation linéaire

Modèle	Transformation	Linéarisation
$y = \theta_0 + \theta_1 \log(x)$	$x' = \log(x)$	$y = \theta_0 + \theta_1 x'$
$y = \theta_0 \exp(\theta_1 x)$	$y' = \log(y)$	$y' = \log(\theta_0) + \theta_1 x$
$y = \frac{x}{(\theta_0 x - \theta_1)}$	$x' = \frac{1}{x}, y' = \frac{1}{y}$	$y' = \theta_0 - \theta_1 x'$
$y = \frac{\exp(\theta_0 + \theta_1 x)}{1 + \exp(\theta_0 + \theta_1 x)}$	$y' = \log\left(\frac{y}{1-y}\right)$	$y' = \theta_0 + \theta_1 x$

Important! Le coefficient de corrélation de Pearson ne quantifie pas une relation de causalité. L'erreur à ne pas faire, c'est de confondre entre la causalité et la corrélation. Afin de nuancer cette différence, considérons deux événements X_1 et X_2 . La causalité s'exprimera comme :

$$X_1 \Rightarrow X_2$$

Qu'on pourra lire, en disant que la cause X_1 engendre (ou cause) l'effet X_2 . Dans cette implication, on note une relation chronologique entre les deux événements. La cause est systématiquement antérieure à l'effet. Le principe de causalité, en physique par exemple, impose une irréversibilité du temps. Cela veut dire que le temps s'écoule de manière linéaire (pas forcément une droite) et on ne pourra pas passer deux fois par le même instant. Autrement dit, on doit suivre l'écoulement du temps. La causalité est vraiment un élément structurant pour la physique. Si le temps était cyclique cela signifie qu'on pourra voyager dans le passé et ainsi rétroagir sur la cause et modifier l'effet, ce qui violerait le principe de causalité. En outre, dans la relation de causalité la cause et l'effet sont définis de façon univoque.

En ce qui concerne la corrélation, les deux événements X_1 et X_2 se produisent simultanément. Déjà à ce stade, on voit bien que la relation antériorité n'existe pas ici. Dans une corrélation, les deux événements (X_1 et X_2) sont engendrés par une cause commune extérieure X_3 . Ainsi, la cause et l'effet ne sont pas définis de façon univoque.

Exemple : on souhaite établir une corrélation entre la taille d'une population et le nombre de médicaments vendus pendant une année. X_1 = Taille de la population et X_2 = nombre de médicaments vendus. La corrélation s'exprime selon le schéma suivant :

$$X_1 \Leftrightarrow X_2$$

Dans cet exemple, la cause commune est X_3 = la maladie. C'est la maladie qui pousse les gens à consommer les médicaments ($X_3 \Rightarrow X_2$). En outre, plus la maladie est contagieuse (une épidémie par exemple) plus le nombre de personne malade est élevé ($X_3 \Rightarrow X_1$). En effet, les deux événements X_1 et X_2 peuvent être corrélés (positivement ou négativement) sans que la variation de l'un n'entraîne la variation de l'autre. Les variations de X_1 et X_2 sont engendrées par la même cause extérieure X_3 , qu'on appelle aussi facteur latent.

V. DIAGNOSTIC DE LA RÉGRESSION

Avant toute modélisation, il est impératif de réaliser un diagnostic de la régression afin d'identifier d'éventuels points atypiques. Dans ce qui suit, nous nous sommes proposés de faire appel à deux outils usuels d'identification de tels points.

A. Effet levier

L'estimation des paramètres d'ajustement est très sensible à la présence de points extrêmes susceptibles d'affecter sérieusement les résultats. L'impact d'un point éloigné de la variable explicative, est donné par le **levier** défini pour chaque observation :

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad \text{avec} \quad \sum_{i=1}^n h_i = p + 1 \quad \text{et} \quad \frac{1}{n} \leq h_i \leq 1$$

Avec $p = 1$ pour la régression linéaire simple. Un point x_i présente un effet levier important si $h_i > 2(p + 1)/n$. L'effet levier apparaît pour des observations ayant des valeurs de la variable explicative loin du centre. Autrement dit, la distance $x_i - \bar{x}$ est élevée. Une observation avec un h_i proche de 1 est une observation avec un levier extrêmement important. Un point ayant un effet levier important est susceptible de **masquer** ou bien de créer une fausse corrélation.

B. Distance de Cook

L'effet levier survient pour des points expérimentaux ayant des variables explicatives loin du barycentre \bar{x} . Un résidu élevé pointe plutôt des valeurs atypiques de la variable à expliquer, ou la réponse. Ces deux diagnostics peuvent se combiner en une seule mesure synthétique définie par la distance de Cook :

$$c_i = \frac{h_i}{p(1 - h_i)^2} \left(\frac{e_i^2}{s_r^2} \right) \quad (32)$$

Une valeur de c_i supérieure à 1, suggère que l'observation i est une mesure influente. Le terme "influent" signifie que l'absence ou la présence de l'observation considérée change substantiellement la valeur des coefficients de régression.

```

clc ; clear all ; close all ;
% Samir KENOUCHE - le 06/09/2019
% METHODE DES MOINDRES CARRES PONDERES

xi = 1:17 ;

yi = [1.04 , 1.98 , 3.38 , 2.94 , 4.72 , 5.12 , 5.84 , 6.05 , 7.80, ...
      9.29 , 9.40 , 9.37 , 11.03 , 14.1 , 13.7 , 13.4 , 14.65] ;

err = [0.3 , 0.4 , 0.7 , 0.5 , 0.9 , 1.05 , 1.15 , 1.23 , 1.28 , ...
      0.9 , 1.7 , 1.60 , 1.75 , 1.45 , 1.27 , 0.8 , 1.10 ] ;

wi = 1./err.^2 ; figure('color',[1 1 1]) ; errorbar(xi,yi,err,'s') ;

numérateur_thetal = sum(wi)*sum(wi.*xi.*yi) - sum(wi.*xi)*sum(wi.*yi) ;
dénominateur = sum(wi)*sum(wi.*(xi.^2)) - (sum(wi.*xi)).^2 ;

thetal = numérateur_thetal/dénominateur ;
numérateur_theta2=sum(wi.*yi)*sum(wi.*(xi.^2))-sum(wi.*xi)*sum(wi.*xi.*yi)
    
```



```
theta_2 = numerateur_theta2/denominateur ;

delta_theta1 = sqrt(sum(wi)/denominateur) ;
delta_theta2 = sqrt(sum(wi.*(xi.^2))/denominateur) ;
dx = 0.01 ; x = 1:dx:20 ; yhat = theta1*x ;
hold on ; plot(x,yhat,'r') ;
```

VI. TRAVAUX PRATIQUES AVEC DES FONCTIONS MATLAB PRÉDÉFINIES

Matlab® dispose de la boîte à outils *Curve Fitting Toolbox* dédiée à l'ajustement. Les fonctions en question sont : `nlinfit`, `fit`, `lsqcurvefit`, `lsqnonlin` et `robustfit`. Nous allons étudier en détail la syntaxe de ces fonctions, à partir de l'exercice ci-dessous :

Exercice 1

- 1) Déterminer le meilleur ajustement des données expérimentales (x_i, y_i) ci-dessous. On donne les deux modèles théoriques :

$$\begin{cases} f_1(x) = \frac{a b x}{(1 + b x)} \\ f_2(x) = a_0 + a_1 \cos(x w) + b_1 \sin(x w) \end{cases} \quad (33)$$

- 2) Tester les fonctions `nlinfit`, `fit`, `lsqcurvefit`, `lsqnonlin`, l'opérateur "backslash" et `robustfit`.
- 3) Afficher, sur la même figure, les données expérimentales et le profil théorique.

Script Matlab®

```
clear all ; close all ; clc ;

% Samir KENOUCHE - le 06/09/2019
% Fonction : lsqcurvefit

xdata = 1:60 ;
ydata = [6.89 1.18e+01 1.91e+01 2.27e+01 2.20e+01 2.21e+01 ...
         2.59e+01 2.80e+01 3.05e+01 2.11e+01 3.50e+01 2.71e+01 ...
         3.04e+01 2.92e+01 3.20e+01 3.12e+01 3.64e+01 3.97e+01 ...
         3.39e+01 3.49e+01 3.38e+01 3.94e+01 3.46e+01 3.79e+01 ...
         3.61e+01 3.77e+01 3.99e+01 4.11e+01 3.59e+01 3.68e+01 ...
         3.79e+01 3.81e+01 4.19e+01 3.89e+01 3.84e+01 3.44e+01 ...
         4.00e+01 4.24e+01 4.37e+01 4.34e+01 3.98e+01 3.96e+01 ...
         4.54e+01 3.49e+01 4.53e+01 4.25e+01 3.96e+01 4.38e+01 ...
         4.03e+01 4.57e+01 4.74e+01 4.73e+01 4.24e+01 4.05e+01 ...
         4.06e+01 4.02e+01 4.14e+01 3.72e+01 4.44e+01 4.49e+01] ;

err = 3.7.*randn(1, numel(xdata)) ;

idelta = [0.01 ; 1] ; % Starting guess
lb = [-inf ; -inf] ; % Lower bound
ub = [+inf ; +inf] ; % Upper bound

[ndelta, resnorm, residual, exitflag, output] = lsqcurvefit(@maFonction, ...
    idelta ,xdata, ydata, lb, ub) ;
```

```
n = 600 ; step = (xdata(end) - xdata(1))/n ; xn = xdata(1) :step: xdata(end) ;
modelFun = (ndelta(1).*ndelta(2).*xn)./(1 + ndelta(2).*xn);

figure('color',[1 1 1]) ;
errorbar(xdata,ydata,err,'o') ; hold on ; plot(xn, modelFun,'r') ;
axis([-3 65 -3 55]) ; xlabel('x'); ylabel('f(x)') ;
legend({'Ajustement', 'Data'},'location','NorthWest') ;
```

La formule mathématique du modèle théorique est écrite séparément dans un fichier *M-file* et sauvegardé sous le nom *maFonction.m* dans le même répertoire que le script appelant :

```
function modelFun = maFonction(idelta, xdata)
modelFun = (idelta(1).*idelta(2).*xdata)./(1 + idelta(2).*xdata);
end
```

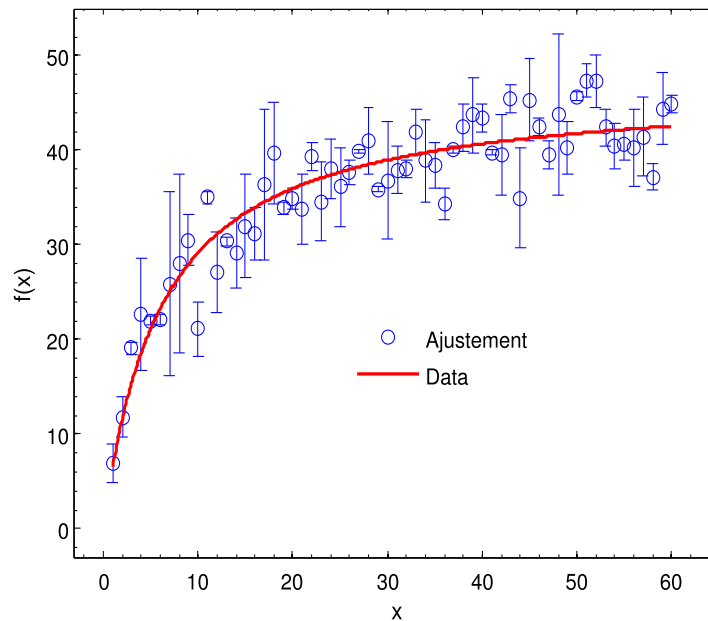


FIGURE 5: Ajustement avec la fonction *lsqcurvefit*

La sortie *ndelta* renvoie les paramètres optimaux, au sens des moindres carrés. La sortie *resnorm* est la moyenne des résidus, calculée sur le nombre de points expérimentaux. L'argument *residual* est le vecteur des résidus, sa longueur est celle du nombre de points expérimentaux. L'argument de sortie *exitflag* = 1 signifie que l'algorithme a convergé vers la solution, une valeur négative indiquera une recherche infructueuse des paramètres optimaux. La sortie *output*, de type *structure*, renvoie le nombre d'itérations effectuées, l'algorithme d'optimisation utilisé, ... etc. La syntaxe usuelle de la fonction *nlinfit* est la suivante :

```
idelta = [0 ; 1] ; % Starting guess

maFonction = @(idelta, xdata) (idelta(1)*idelta(2)*xdata)./(1 + idelta(2)*
    xdata) ;

[ndelta, residual, Jacob, COVB, mse] = nlinfit(xdata, ydata, maFonction,
    idelta)
```

```

modelFun = (ndelta(1)*ndelta(2)*xdata)./(1 + ndelta(2)*xdata) ;
plot(xdata, ydata, 'o') ; hold on ; plot(xdata, modelFun, 'r') ;

interPara = nlparci(ndelta, residual, 'covar', COVB) ;
    
```

L'argument `ndelta` est le vecteur des paramètres optimaux. La sortie `r` est le vecteur des résidus. La sortie `J` est le *Jacobien* de la fonction de régression. `COB` est la matrice de *covariance* des paramètres. À partir des éléments diagonaux de cette matrice, on peut calculer l'erreur commise sur chaque paramètre. Les éléments non-diagonaux servent à calculer le coefficient de corrélation. Le dernier argument de sortie `mse` est la moyenne du carré des écarts, ou des distances verticales. Le RMSE est obtenu à partir de la racine carrée de `mse`.

La fonction `nlparci` renvoie l'intervalle de confiance des paramètres estimés. Par défaut, cette fonction utilise une probabilité de 0.95 (\pm deux fois l'écart-type) de trouver le paramètre dans l'intervalle de confiance correspondant. Le nombre de ligne de la matrice `interPara` est égale au nombre de paramètres à estimer. Il est possible également d'obtenir l'intervalle de confiance sur les réponses du modèle de régression, via la fonction `nlpredci` :

```

idelta = [0 ; 1] ; % Starting guess

maFonction = @(idelta, xdata) (idelta(1)*idelta(2)*xdata)./(1 + idelta(2)*
    xdata) ;
[ndelta, residual, Jacob, COVB, mse] = nlinfit(xdata, ydata, maFonction, ...
    idelta)
[ypred, ecart] = nlpredci(maFonction, xdata, ndelta, residual, 'covar', COVB)

plot(xdata, ypred + ecart, 'g') ; hold on ;
plot(xdata, ypred - ecart, 'g')
    
```

La fonction `nlinfit` peut aussi être utilisée avec la syntaxe :

```

idelta = [0 ; 1] ; % Starting guess
% chargement des xdata et ydata
maFonction = @(idelta, xdata) (idelta(1)*idelta(2)*xdata)./(1 + idelta(2)*
    xdata);

coefEsts = nlinfit(xdata, ydata, maFonction, idelta) ; n = 600 ;
step = (xdata(end) - xdata(1))/n ; xn = xdata(1) :step: xdata(end) ;
line(xn, maFonction(coefEsts, xn), 'Color', 'r')
    
```

La syntaxe de la fonction `lsqnonlin` est très similaire à celle de `lsqcurvefit` :

```

idelta = [60 ; 0.1] ; % Starting guess
lb = [10 ; 0.5] ; % Lower bound
ub = [1e+06 ; 1e+04] ; % Upper bound

opts = optimset('Jacobian', 'on') ;
[ndelta, resnorm, residual, exitflag, output] = lsqnonlin(@maFonction, idelta, ...
    xdata, lb, ub, opts) ;
    
```

```
modelFun = (ndelta(1)*ndelta(2)*xdata)./(1 + ndelta(2)*xdata) ;
plot(xdata, ydata, 'o') ; hold on ; plot(xdata, modelFun, 'r') ;
```

Les algorithmes de *Gauss-Newton* et de *Levenberg-Marquardt* sont accessibles via le champ d'optimisation `optimset`, selon la syntaxe :

```
opts = optimset('Jacobian', 'on', 'LevenbergMarquardt', 'on') ;
[ndelta, resnorm, residual, exitflag, output] = lsqnonlin(@maFonction, idelta, lb,
ub, opts) ;
```

Pour la méthode *Gauss-Newton*, on écrira `opts = optimset('gn', 'on')`. La méthode de la *descente de gradient* est accessible via `opts = optimset('HessUpdate', 'steepdesc')`. Les données expérimentales peuvent également être ajustées avec la fonction `fit`, suivant la syntaxe :

```
idelta = [20 ; 0.1] ; % Starting guess
lb = [-inf ; -inf] ; % Lower bound
ub = [+inf ; +inf] ; % Upper bound

s = fitoptions('Method', 'NonlinearLeastSquares', ...
'Lower', idelta, 'Upper', ub, 'Startpoint', lb) ;

fun = fittype('a*b*x./(1 + b*x)', 'options', s) ;
[modelFun, stat] = fit(xdata, ydata, fun) ;
plot(xdata, ydata, 'o') ; hold on ; plot(modelFun, 'r') ;
```

Les sorties renvoyées sont :

```
modelFun =
    General model:
    modelFun(x) = a*b*x./(1 + b*x)
    Coefficients (with 95 % confidence bounds):
    a =          32.71 (27.73, 37.69)
    b =              1 (fixed at bound)

stat      =
    sse: 375.1648
    rsquare: 0.7631
    dfe: 10
    adjrsquare: 0.7631
    rmse: 6.1251
```

Le champ de structure `fitoptions` indique les modalités d'optimisation. Sa syntaxe générale est :

```
s = fitoptions('Method', method, 'PropertyName1', PropertyValue1,
'PropertyName2', PropertyValue2, ...)
```

L'argument `method` désigne la méthode d'optimisation considérée qui sont : `'LinearInterpolant'` (interpolation linéaire), `'PchipInterpolant'` (interpolation d'*Hermite*), `'CubicSplineInterpolant'` (interpolation par spline cubique), `'SmoothingSpline'` (moyenne par spline) et `'NonlinearLeastSquares'` (moindres carrés). Les arguments `Lower` et `Upper` sont respectivement les bornes inférieure et supérieure des paramètres à estimer. L'argument `Startpoint` correspond aux valeurs initiales données aux paramètres. Les autres champs d'optimisation sont accessibles, en affichant la sortie `s`, qui renvoie :

```
s =

    Normalize: 'off'
    Exclude: []
    Weights: []
    Method: 'NonlinearLeastSquares'
    Robust: 'Off'
    StartPoint: [-Inf -Inf]
    Lower: [20 0.1000]
    Upper: [Inf Inf]
    Algorithm: 'Trust-Region'
    DiffMinChange: 1.0000e-08
    DiffMaxChange: 0.1000
    Display: 'Notify'
    MaxFunEvals: 600
    MaxIter: 400
    TolFun: 1.0000e-06
    TolX: 1.0000e-06
```

Un argument ayant pour valeur l'ensemble vide [], signifie que la propriété correspondante est inactive. Les propriétés affichées avec les chaînes de caractères, sont celles indiquées dans le champ de structure `fitoptions`. Les autres propriétés, par exemple `MaxFunEvals`, `TolFun`, `TolX`, ...etc, ont des valeurs par défaut. L'argument de sortie `modelFun` est le modèle de régression avec les paramètres optimaux. Chaque coefficient (paramètre) est affiché avec son intervalle de confiance. La sortie `stat` contient les valeurs des outils statistiques permettant l'estimation de la qualité de l'ajustement. On retrouve notamment, l'erreur quadratique moyenne (`rmse`), le coefficient de détermination (`rsquare`) et le coefficient de détermination ajusté (`adjrsquare`). L'argument `dfe` est le nombre de degré de liberté considéré. On peut afficher également l'*intervalle de confiance* de l'ajustement par le biais de la commande `predint`, selon la syntaxe :

```
vall = predint(modelFun, xdata, 0.95, 'observation', 'on') ;
hold on ; plot(xdata, vall, 'm---') ;
```

Il est possible, en outre, de calculer l'intégrale, la première et la seconde dérivée du modèle d'ajustement. La syntaxe est la suivante :

```
plot(modelFun, xdata, ydata, {'int', 'deriv1', 'deriv2'})
```

Cette intégrale et ces dérivées peuvent être obtenues aussi avec les instructions :

```
[der1, der2] = differentiate(modelFun, xdata) ;
int = integrate(modelFun, xdata, xdata(1)) ;
```

On a aussi la possibilité d'ajuster sans tenir compte de points aberrants, (*outliers* en anglais), en utilisant les instructions suivantes :

```
outliers = excludedata(xdata,ydata,'range',[-ymin ymax]) ;
modelFun = fit(xdata', ydata', fun, 'startpoint', [20 ; 0.1], 'exclude', outliers)
```

Les points aberrants sont inclus dans l'intervalle [-ymin ymax], voir aussi `coeffnames(modelFun)`, `formula(modelFun)` et `islinear(modelFun)`. La liste complète des commandes Matlab relative à l'ajustement par la fonction `fit` est accessible via `: methods(fun)`. Désormais, on réalisera le même ajustement avec l'opérateur `backslash \`

```
clc ; clear all ;
% Samir KENOUCHE - 1e 06/09/2019

syms a b x
fx = a*b*x/(1 + b*x) ; dif1 = diff(fx, a) ; dif2 = diff(fx, b) ;

dif1Val = subs(dif1, {'a', 'b', 'x'}, {1 1 xData'}) ;
dif2Val = subs(dif2, {'a', 'b', 'x'}, {1 1 xData'}) ;

xn = [dif1Val dif2Val] ;

coeffs = xn \ yData' ; % OPERATEUR backslash : "\"
yn = [dif1Val dif2Val]*coeffs ;
plot(xData, yData, 'o') ; hold on ; plot(xData, yn, 'r')
```

Avec la fonction `robustfit`, on utilisera la syntaxe :

```
clc ; clear all ; close all ;
% Samir KENOUCHE - 1e 06/09/2019

syms a b x
fx = a*b*x/(1 + b*x) ; dif1 = diff(fx, a) ; dif2 = diff(fx, b) ;

dif1Val = subs(dif1, {'a', 'b', 'x'}, {1 1 xData}) ;
dif2Val = subs(dif2, {'a', 'b', 'x'}, {1 1 xData}) ;

xn = [dif1Val ; dif2Val] ;
[coeffs, stats] = robustfit(xn', yData') ;
modelFun = coeffs(1)*coeffs(2).*xData./(1 + coeffs(1).* xData) ;
plot(xData, yData, 'o') ; hold on ; plot(xData, modelFun, 'r') ;
```

La fonction `robustfit` détermine les paramètres d'ajustement, selon la méthode des *moindres carrés pondérés*, soit :

$$S(\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_i) = \sum_i^n w_i (y_i - f(x_i; \theta_i))^2 \tag{34}$$

Le vecteur w_i donne la contribution (ou le poids) de chaque mesure, il est donné en fonction des résidus. Ainsi, chaque point expérimental est pondéré via la quantité :

$$r = \frac{(y_i - \hat{y}_i)}{tune \ s \ \sqrt{1 - h}} \tag{35}$$

Avec, h est une fonction décrivant la contribution de chaque mesure expérimentale sur le modèle de régression. La constante $tune$, dont les valeurs par défaut sont énumérées ci-dessous, permet de moduler (augmenter ou diminuer) le poids de chaque mesure. Le paramètre s est donné par :

$$s = \frac{\text{median}(|r|)}{0.6745} \tag{36}$$

Ci-dessous, les différentes méthodes de pondération de la fonction `robustfit`

'andrews'	w = (abs(r)<pi) .* sin(r) ./ r	Default Tuning Constant	1.339
'bisquare'	w = (abs(r)<1) .* (1-r.^2).^2	Default Tuning Constant	4.685
'cauchy'	w = 1 ./ (1 + r.^2)	Default Tuning Constant	2.385
'fair'	w = 1 ./ (1 + abs(r))	Default Tuning Constant	1.400
'huber'	w = 1 ./ max(1, abs(r))	Default Tuning Constant	1.345
'logistic'	w = tanh(r) ./ r	Default Tuning Constant	1.205
'talwar'	w = 1 * (abs(r)<1)	Default Tuning Constant	2.795
'welsch'	w = exp(-(r.^2))	Default Tuning Constant	2.985

Tous ces champs d'optimisation sont indiqués avec la syntaxe générale suivante :

```
[coeffs, stats] = robustfit(x, y, wfun, tune, valTune)
```

Les arguments de sortie x et y sont respectivement la matrice *Jacobienne* et le vecteur des réponses observées. Le nombre de ligne de x est égale au nombre de point expérimentaux et le nombre de colonne est égale au nombre de paramètre à estimer. L'argument `wfun` indique la méthode ('logistic', 'cauchy', ..., etc) de pondération utilisée. Une valeur (`valTune`) faible de la constante `tune` tendra à réduire substantiellement la contribution des mesures à large résidu, et inversement. Pour le modèle théorique décrit par la relation $f_2(x)$, on optera pour les commandes `fit` et `fittype` selon le script suivant :

```
clc ; clear all ; close all ;
% Samir KENOUCHE - le 06/09/2019

xdata = linspace(0, 110, 219) ;
ydata = [2.98 2.71 2.61 1.97 2.42 0.75 0.79 0.69 0.86 1.28 0.99 ...
0.76 1.93 1.28 0.27 1.51 1.36 3.19 2.43 2.33 2.73 3.00 2.89 ...
3.06 2.71 3.26 2.19 1.00 1.27 1.18 1.43 1.03 0.98 0.38 0.66 ...
0.89 0.58 0.63 1.67 3.01 3.09 2.70 2.82 2.97 2.86 2.92 2.45 ...
1.79 2.07 0.23 3.05 1.03 1.52 0.82 1.01 0.83 1.67 1.49 1.65 ...
1.70 2.40 2.74 2.99 2.93 3.06 2.86 3.12 3.01 3.41 2.16 2.40 ...
0.88 0.91 1.07 0.42 0.92 1.02 0.85 0.68 1.34 1.83 2.02 2.64 ...
```

```

3.20 2.29 0.93 2.97 3.06 2.82 2.27 2.35 2.55 3.05 1.61 1.73 ...
2.10 1.06 0.89 1.07 0.75 1.82 1.01 2.96 1.36 2.27 2.74 2.80 ...
2.89 3.15 3.01 2.64 1.88 3.15 3.29 0.90 1.99 1.20 0.40 0.66 ...
1.07 0.72 1.35 1.74 1.19 0.07 2.04 3.30 2.22 3.63 2.67 3.17 ...
2.91 2.70 2.69 3.11 3.19 0.86 0.67 1.20 0.45 1.08 0.83 1.08 ...
0.98 2.20 1.34 1.65 2.81 3.00 2.42 3.11 2.86 3.03 2.85 2.39 ...
3.10 2.50 2.80 1.50 2.18 1.62 1.47 0.84 1.06 1.04 1.33 1.89 ...
1.15 1.92 2.89 2.00 2.16 3.04 2.65 3.04 2.96 3.00 2.61 1.47 ...
2.81 2.46 2.13 0.85 1.26 0.89 0.98 1.06 0.36 1.15 0.83 1.06 ...
1.50 2.26 3.14 2.91 2.73 2.00 2.89 2.04 3.03 3.01 2.06 0.96 ...
2.17 0.88 1.78 1.24 0.97 1.46 0.06 1.53 1.29 1.54 2.55 2.53 ...
1.69 2.56 2.80 2.99] ;

err = 0.2*randn(1, numel(ydata)) ; % vecteur des erreurs

idelta = [1 ; 1 ; 0.01 ; 0.49] ; % Starting guess
lb = [0.1; 0.1 ; -1 ; 0.3] ; % Lower bound
ub = [+inf ; +inf ; +inf ; 0.6] ; % Upper bound
s = fitoptions('Method','NonlinearLeastSquares',...
    'Lower',idelta,'Upper',ub,'Startpoint',lb) ;

fun = fittype('a0 + a1*cos(x*w) + b1*sin(x*w)','options',s) ;
[modelFun, stat] = fit(xdata', ydata', fun) ;
figure('color',[1 1 1]) ;
errorbar(xdata, ydata, err, '.'); hold on ; plot(modelFun,'r') ;
xlabel('x') ; ylabel('f(x)') ; legend('data','fitted curve') ;

```

Voici le graphique généré par ce script :

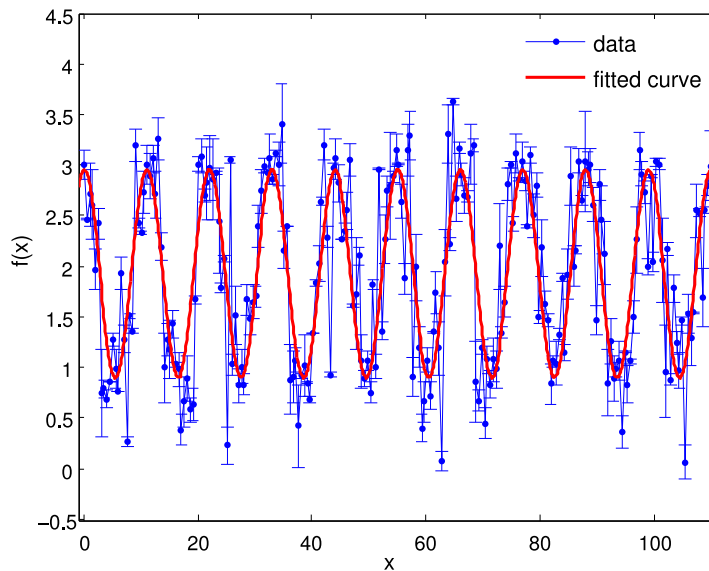


FIGURE 6: Ajustement avec la fonction *fit*

Les sorties `modelFun` et `stat` renvoient les valeurs des paramètres d'ajustement du modèle et les différents outils statistiques évaluant la qualité de l'ajustement.

```
>> modelFun

General model:
modelFun(x) = a0 + a1*cos(x*w) + b1*sin(x*w)
Coefficients (with 95 % confidence bounds):
a0 =      1.9200  (1.848, 1.993)
a1 =      1.0160  (0.9141, 1.118)
b1 =      0.0100  (fixed at bound)
w =      0.5724  (0.5708, 0.574)

>> stat

      sse: 63.5619  % SOMME DES CARRES DES ECARTS
rsquare: 0.6412   % COEFFICIENT DE CORRELATION
      dfe: 216     % NOMBRE DE DEGRES DE LIBERTE
adjrsquare: 0.6378 % COEFFICIENT DE CORRELATION ADJUSTE
      rmse: 0.5425 % RACINE CARREE DES ECARTS MOYENS

% La sortie fun renvoie :
>> fun =

General model:
fun(a0,a1,b1,w,x) = a0 + a1*cos(x*w) + b1*sin(x*w)
```

QUANTILES DE LA LOI DE STUDENT

On rappelle :

$$\mathbb{P}(T > t_{n-2,\alpha}) = 1 - \mathbb{P}(T < t_{n-2,\alpha}) = \alpha$$

Niveau de confiance

Table		Quantiles de la loi de Student								
	0,6	0,7	0,8	0,9	0,95	0,975	0,99	0,995	0,999	0,9995
1	0,3249	0,7265	1,3764	3,0777	6,3137	12,7062	31,8210	63,6559	318,2888	636,5776
2	0,2887	0,6172	1,0607	1,8856	2,9200	4,3027	6,9645	9,9250	22,3285	31,5998
3	0,2767	0,5844	0,9785	1,6377	2,3534	3,1824	4,5407	5,8408	10,2143	12,9244
4	0,2707	0,5686	0,9410	1,5332	2,1318	2,7765	3,7469	4,6041	7,1729	8,6101
5	0,2672	0,5594	0,9195	1,4759	2,0150	2,5706	3,3649	4,0321	5,8935	6,8685
6	0,2648	0,5534	0,9057	1,4398	1,9432	2,4469	3,1427	3,7074	5,2075	5,9587
7	0,2632	0,5491	0,8960	1,4149	1,8946	2,3646	2,9979	3,4995	4,7853	5,4081
8	0,2619	0,5459	0,8889	1,3968	1,8595	2,3060	2,8965	3,3554	4,5008	5,0414
9	0,2610	0,5435	0,8834	1,3830	1,8331	2,2622	2,8214	3,2498	4,2969	4,7809
10	0,2602	0,5415	0,8791	1,3722	1,8125	2,2281	2,7638	3,1693	4,1437	4,5868
11	0,2596	0,5399	0,8755	1,3634	1,7959	2,2010	2,7181	3,1058	4,0248	4,4369
12	0,2590	0,5386	0,8726	1,3562	1,7823	2,1788	2,6810	3,0545	3,9296	4,3178
13	0,2586	0,5375	0,8702	1,3502	1,7709	2,1604	2,6503	3,0123	3,8520	4,2209
14	0,2582	0,5366	0,8681	1,3450	1,7613	2,1448	2,6245	2,9768	3,7874	4,1403
15	0,2579	0,5357	0,8662	1,3406	1,7531	2,1315	2,6025	2,9467	3,7329	4,0728
16	0,2576	0,5350	0,8647	1,3368	1,7459	2,1199	2,5835	2,9208	3,6861	4,0149
17	0,2573	0,5344	0,8633	1,3334	1,7396	2,1098	2,5669	2,8982	3,6458	3,9651
18	0,2571	0,5338	0,8620	1,3304	1,7341	2,1009	2,5524	2,8784	3,6105	3,9217
19	0,2569	0,5333	0,8610	1,3277	1,7291	2,0930	2,5395	2,8609	3,5793	3,8833
20	0,2567	0,5329	0,8600	1,3253	1,7247	2,0860	2,5280	2,8453	3,5518	3,8496
21	0,2566	0,5325	0,8591	1,3232	1,7207	2,0796	2,5176	2,8314	3,5271	3,8193
22	0,2564	0,5321	0,8583	1,3212	1,7171	2,0739	2,5083	2,8188	3,5050	3,7922
23	0,2563	0,5317	0,8575	1,3195	1,7139	2,0687	2,4999	2,8073	3,4850	3,7676
24	0,2562	0,5314	0,8569	1,3178	1,7109	2,0639	2,4922	2,7970	3,4668	3,7454
25	0,2561	0,5312	0,8562	1,3163	1,7081	2,0595	2,4851	2,7874	3,4502	3,7251
26	0,2560	0,5309	0,8557	1,3150	1,7056	2,0555	2,4786	2,7787	3,4350	3,7067
27	0,2559	0,5306	0,8551	1,3137	1,7033	2,0518	2,4727	2,7707	3,4210	3,6895
28	0,2558	0,5304	0,8546	1,3125	1,7011	2,0484	2,4671	2,7633	3,4082	3,6739
29	0,2557	0,5302	0,8542	1,3114	1,6991	2,0452	2,4620	2,7564	3,3963	3,6595
30	0,2556	0,5300	0,8538	1,3104	1,6973	2,0423	2,4573	2,7500	3,3852	3,6460
31	0,2555	0,5298	0,8534	1,3095	1,6955	2,0395	2,4528	2,7440	3,3749	3,6335
32	0,2555	0,5297	0,8530	1,3086	1,6939	2,0369	2,4487	2,7385	3,3653	3,6218
33	0,2554	0,5295	0,8526	1,3077	1,6924	2,0345	2,4448	2,7333	3,3563	3,6109
34	0,2553	0,5294	0,8523	1,3070	1,6909	2,0322	2,4411	2,7284	3,3480	3,6007
35	0,2553	0,5292	0,8520	1,3062	1,6896	2,0301	2,4377	2,7238	3,3400	3,5911
36	0,2552	0,5291	0,8517	1,3055	1,6883	2,0281	2,4345	2,7195	3,3326	3,5821
37	0,2552	0,5289	0,8514	1,3049	1,6871	2,0262	2,4314	2,7154	3,3256	3,5737
38	0,2551	0,5288	0,8512	1,3042	1,6860	2,0244	2,4286	2,7116	3,3190	3,5657
39	0,2551	0,5287	0,8509	1,3036	1,6849	2,0227	2,4258	2,7079	3,3127	3,5581
40	0,2550	0,5286	0,8507	1,3031	1,6839	2,0211	2,4233	2,7045	3,3069	3,5510
50	0,2547	0,5278	0,8489	1,2987	1,6759	2,0086	2,4033	2,6778	3,2614	3,4960
60	0,2545	0,5272	0,8477	1,2958	1,6706	2,0003	2,3901	2,6603	3,2317	3,4602
70	0,2543	0,5268	0,8468	1,2938	1,6669	1,9944	2,3808	2,6479	3,2108	3,4350
80	0,2542	0,5265	0,8461	1,2922	1,6641	1,9901	2,3739	2,6387	3,1952	3,4164
90	0,2541	0,5263	0,8456	1,2910	1,6620	1,9867	2,3685	2,6316	3,1832	3,4019
100	0,2540	0,5261	0,8452	1,2901	1,6602	1,9840	2,3642	2,6259	3,1738	3,3905
200	0,2537	0,5252	0,8434	1,2858	1,6525	1,9719	2,3451	2,6006	3,1315	3,3398
500	0,2535	0,5247	0,8423	1,2832	1,6479	1,9647	2,3338	2,5857	3,1066	3,3101
∞	0,2533	0,5244	0,8416	1,2816	1,6449	1,9600	2,3264	2,5758	3,0902	3,2905

Degrés de liberté

Bibliographie

- [1] Cooper J., *A Matlab Companion for Multivariable Calculus*. Harcourt/Academic Press, 2001. USA.
- [2] Jędrzejewski F., *Introduction aux méthodes numériques*, Deuxième édition, Springer-Verlag, 2005. France.
- [3] Filbet F., *Analyse numérique - Algorithmes et étude mathématique*. Dunod, 2009.
- [4] Ciarlet P. G., *Introduction à l'analyse numérique matricielle et à l'optimisation - cours et exercices corrigés. Mathématiques appliquées pour la maîtrise*. Dunod, 1998.
- [5] Lascaux P., Théodor R., *Analyse numérique matricielle appliquée à l'art de l'ingénieur. Méthodes itératives*. Dunod, 2000.
- [6] Dion J. G., Gaudet R., *Méthodes d'Analyse Numérique : de la théorie à l'application*. MODULO, 1996.
- [7] Atkinson K., *An Introduction to Numerical Analysis*. 2nd edition. John Wiley & Sons Inc., New York. 1989.
- [8] Axelsson O., *Iterative Solution Methods*. Cambridge University Press. Cambridge, 1994.
- [9] Lambert J., *Numerical Methods for Ordinary Differential Systems*. John Wiley and Sons, Chichester, 1991.
- [10] Crouzeix M., Mignot A., *Analyse numérique des équations différentielles*. Masson, 1984.
- [11] Björck A., *Numerical Methods for Least Squares Problems*. SIAM. Philadelphia, 1996.
- [12] Veysseyre R., *Statistique et probabilités pour l'ingénieur*. 2^{ème} édition Dunod, Paris. 2006.
- [13] Protassov K., *Analyse statistique des données expérimentales*. EDP Sciences, Grenoble. 2002.