République Algérienne Démocratique et Populaire.
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.
Université Mouloud Mammeri, Tizi-Ouzou.
Faculté des Sciences.
Département de Mathématiques.

Spécialité : Mathématiques
Option : Probabilités et statistique
Mémoire de fin de cycle intitulé :

Modèles de régression spline et application au Machine learning

---

Réalisé par :

Ait Ramdane Thanina

Encadré par :

M^{r} **Fellag Hocine**

Soutenu devant le jury d'examen composé de :

M^{me} **Atil Lynda,**	Maître de conférence A, UMMTO,	Présidente.
M^{r} **Fellag Hocine,**	Professeur, UMMTO,	Rapporteur.
M^{me} **Belkacem Cherifa,**	Maître de conférence B, UMMTO,	Examinatrice.

Soutenu le : 25/09/2024

Table des matières

Dédicace

Je dédie ce modeste travail à ma chère maman ***Djoudi.T*** *pour son amour inconditionnel, son soutien et ses sacrifices. Tu as cru en moi, même quand moi je doutais. Ton encouragement m'a donné la force de persévérer.*

À mes chers frères ***M. Salah***, ***Juba*** *et à ma chère grand-mère* ***Ben Makhelouf.F****, pour leur soutien constant et leur compréhension pendant les périodes d'études intenses. Votre présence et vos mots d'encouragement ont été un pilier essentiel dans la réalisation de ce mémoire.*

À mes amies (***Imane, Lydia, Souhila, Amira, Kenza, Dihia et Sadia****), pour leur camaraderie, leurs encouragements et les moments de joie partagés. Votre présence a rendu ce parcours plus agréable et supportable.*

Même si la vie a pris mon père qui rêvait de ce jour, Dieu m'a récompensé avec son ami ***Almas Ali*** *qui m'a toujours soutenu à chaque étape de ma vie.*

Enfin, à tous ceux qui croient en moi et m'ont soutenu tout au long de cette aventure académique, ce mémoire est aussi le vôtre, car chacun de vous a contribué à sa manière à son accomplissement.

A.R.Thanina

Remerciements

La réalisation de ce mémoire a été possible grâce au Dieu et à plusieurs personnes à qui je voudrais témoigner toute ma gratitude.

Je tiens tout d'abord à remercier mon encadrant ***Monsieur Fellag Hocine****, pour sa grande disponibilité, sa patience et ses précieux conseils. Vous avez partagé votre savoir avec passion et générosité, ce qui largement contribué à ma réussite.*

Mes vifs remerciements vont également à M^{me} ***Atil Lynda*** *pour l'honneur qu'elle me fait en présidant le jury de ce mémoire ainsi qu'à* M^{me} ***Belkacem Cherifa****, qui a accepté d'examiner ce travail.*

Je souhaite également exprimer ma reconnaissance envers mes professeurs pour la qualité de leur enseignement et les outils indispensables qu'ils m'ont fournis pour réussir mes études universitaires. Je n'oublie pas mon enseignante M^{me} ***Zennouche Zina****, qui a été la première à me faire découvrir les mathématiques.*

Enfin, je remercie toutes personnes qui ont contribué de prés ou de loin à la réalisation de cet humble travail.

Introduction générale

L'intelligence Artificielle (IA) a profondément transformé la manière dont nous comprenons et exploitons les données. En utilisant des algorithmes avancés, l'IA permet de simuler des processus intellectuels tels que l'apprentissage, la prise de décision et la reconnaissance de modèles. Ce domaine qui englobe le *machine learning*, la vision par ordinateur et bien d'autres sous-domaines est devenu essentiel dans de nombreux secteurs comme la santé, la finance et l'éducation, etc.

L'analyse des données qui est un autre pilier fondamental de la science moderne ; désigne un processus consistant à examiner, transformer et modéliser des données, afin de découvrir des informations utiles, d'en tirer des conclusions et de soutenir la prise de décisions. Parmi ces nombreuses techniques de modélisation, *les modèles de régression spline* jouent un rôle central. Ces modèles sont utilisés pour établir des relations entre des variables indépendantes et une variable dépendante. Cependant, les méthodes classiques de régression comme la régression linéaire, peuvent se révéler insuffisantes lorsque les relations entre les variables sont complexes et non-linéaires. C'est dans ce contexte que les modèles de régression spline trouvent leur utilité.

L'application des splines en machine learning offre des opportunités favorables notamment, pour les problèmes où la structure des données est complexe et difficile à modéliser par des méthodes linéaires ou paramétriques. Dans le contexte de machine learning, les splines peuvent être intégrées dans des algorithmes prédictifs, en particulier lorsque les données présentent des non-linéarités ou des discontinuités.

Ce mémoire est basé sur l'étude des méthodes de régression spline dans le cadre du machine learning. Il est structuré en trois chapitres.

Le premier chapitre abordera les modèles de régression traditionnels en mettant l'accent sur leurs principes fondamentaux et leurs applications. Le second chapitre s'intéressera aux splines en détaillant leur construction, leurs propriétés et leur utilité dans les modèles non paramétriques. Enfin, le troisième chapitre présentera une application en machine learning illustrant l'efficacité des splines dans des scénarios réels.
Nous terminerons par une conclusion générale pour synthétiser notre travail.

Chapitre 1

Modèles de régression

Le terme "régression" en statistique a été introduit par le scientifique britannique Sir Francis Galton en 1886. À cette époque, Galton réalisait des recherches sur l'hérédité, en particulier sur la manière dont la taille des individus variait en fonction de celle de leurs parents. Les résultats de ses études l'ont amené à développer ce qu'il a appelé la "théorie de la régression vers la moyenne."

La régression est l'une des méthodes les plus utilisées en statistique, elle vise à étudier et à modéliser la relation entre une ou plusieurs variables indépendantes dites *variables explicatives* notées X et une variable dépendante Y, appelée *variable à expliquer*(ou *réponse*).

Il existe différents types de régression, paramétriques et non paramétriques , chaque catégorie possède des critères spécifiques pour son application. Dans ce chapitre, nous avons introduit trois types de modèles de regression les plus couramment utilisés : La régression linéaire, la régression polynomiale(sont des modèles de régression paramétrique) et la régression par noyau (régression non paramétrique) .

I. Régression paramétrique

La régression paramétrique se distingue par son approche explicite de la modélisation des relations entre les variables. Elle repose sur des hypothèses relatives à la forme fonctionnelle de la relation, souvent linéaire, ce qui facilite l'interprétation des résultats et l'application de tests statistiques.

D'un point de vue mathématique, cette méthode peut être formalisée de la manière suivante : Soit Y la variable dépendante et $X_1, X_2, \ldots, X_p$ les variables indépendantes. La régression paramétrique suppose que la relation entre Y et $X_1, X_2, \ldots, X_p$ peut être décrite par le modèle suivant :

$$Y = f(X_1, X_2, \ldots, X_p; \beta) + \epsilon$$

où :

- $f(\cdot)$ est une fonction paramétrique qui décrit la relation entre les variables.
- β représente les paramètres du modèle à estimer.

- ϵ est un terme d'erreur qui capture l'incertitude ou le bruit dans la relation entre Y et les variables explicatives X.

1.1 Régression linéaire

La régression linéaire est une méthode de modélisation de la relation entre une variable dépendante et une ou plusieurs variables indépendantes, en supposant une relation linéaire entre elles. Elle cherche à minimiser la différence entre les valeurs observées de la variable dépendante et les valeurs prédites par le modèle.
La régression linéaire simple repose sur l'utilisation d'une seule variable explicative, tandis que la régression linéaire multiple implique l'utilisation de plusieurs variables explicatives.

1.1.1 Régression linéaire simple

Présentation du modèle

Soit $(x_1, y_1), ..., (x_n, y_n)$ un échantillon de n observations. Le modèle de régression simple s'écrit sous la forme suivante :

$$Y = f(X) + \epsilon \tag{1.1}$$

avec f est une fonction à déterminer, ϵ une variable aléatoire(erreur) de loi normale $N(0, \sigma^2)$ et X une variable déterministe(non aléatoire). Dans le cas linéaire, le modèle s'écrit sous la forme :

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i \quad \forall i = 1, ..., n. \tag{1.2}$$

où β_0 et β_1 sont des constantes réelles inconnues à estimer et ϵ_i indépendantes et identiquement distribués $(i.i.d)$ avec

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

Dans cette section, on suppose que les ϵ_i gaussiens et donc Y_i est de loi normale $N(\beta_0 + \beta_1 x_i, \sigma^2)$ pour tout $i = 1, ..., n$. Dans la suite, nous notons Var la matrice de variance covariance.

Estimation des paramètres du modèle de régression

Les estimateurs du maximum de vraisemblance (MV)

Considérons les variables Y_i qui suivent une distribution gaussienne pour tout $i = 1, .., n$ de fonctions de densité :

$$f_{Y_i}(y_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ \frac{-1}{2\sigma^2}(y_i - f(x_i))^2 \right\}.$$

La fonction de vraisemblance associée aux observations (x_1, y_1),...,(x_n, y_n) est comme suit :

$$\begin{aligned} L(\beta_0, \beta_1, \sigma^2, (x_1, y_1), .., (x_n, y_n)) &= \prod_{i=1}^{n} f_{Y_i}(y_i) \\ &= \prod_{i=1}^{n} \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ \frac{-1}{2\sigma^2}(y_i - (\beta_0 + \beta_1 x_i))^2 \right\} \right) \\ &= \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left\{ \frac{-1}{2\sigma^2} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2 \right\}. \end{aligned}$$

La log-vraisemblance est :

$$\begin{aligned} l(\beta_0, \beta_1, \sigma^2, (x_1, y_1), .., (x_n, y_n)) &= \log L(\beta_0, \beta_1, \sigma^2, (x_1, y_1), .., (x_n, y_n)) \\ &= -\frac{n}{2}(\log(2\pi) + \log \sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2. \end{aligned}$$

- Calculons l'estimateur du maximum de vraisemblance pour β_0 :
  En appliquant la dérivation par rapport à β_0, on obtient :

$$\frac{\partial l}{\partial \beta_0} = \frac{1}{\sigma^2} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i).$$

  Ensuite, nous égalons cette dérivée à zéro et résolvons pour β_0 :

$$\begin{aligned} \frac{\partial l}{\partial \beta_0} = 0 &\Leftrightarrow \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i)) = 0 \\ &\Leftrightarrow \sum_{i=1}^{n} y_i - n\beta_0 - \beta_1 \sum_{i=1}^{n} x_i = 0 \\ &\Leftrightarrow n\beta_0 = \sum_{i=1}^{n} y_i - \beta_1 \sum_{i=1}^{n} x_i \\ &\Leftrightarrow \beta_0 = \frac{1}{n} \sum_{i=1}^{n} y_i - \frac{\beta_1}{n} \sum_{i=1}^{n} x_i = \overline{y} - \beta_1 \overline{x}. \end{aligned}$$

  La dérivée seconde par rapport à β_0 est donnée par :

$$\frac{\partial^2 l}{\partial \beta_0^2} = -\frac{n}{\sigma^2} < 0.$$

  Donc, $\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x}$ est un maximum de vraisemblance.

  Par le même raisonnement, on trouve $\hat{\beta}_1$ et $\hat{\sigma^2}$:

- L'estimateur du maximum de vraisemblance pour β_1 est :

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}.$$

- L'estimateur du maximum de vraisemblance pour σ^2 est :

$$\hat{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n}[y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)]^2.$$

Remarque : Les estimateurs du maximum de vraisemblance $\hat{\beta}_0$ et $\hat{\beta}_1$ sont les mêmes que les estimateurs des moindres carrés. Cela découle du fait que le couple $(\hat{\beta}_0, \hat{\beta}_1)$ représente la solution qui minimise la somme des carrés des résidus, définie par :

$$\sum_{i=1}^{n}[y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)]^2.$$

En effet, minimiser cette quantité est équivalant à maximiser la fonction l donnée ci-dessus.
La différence $y_i - (\beta_0 + \beta_1 x_i)$ est appelée résidu, souvent notée par ϵ_i, elle représente l'écart entre la valeur observée y_i et la valeur estimée par le modèle $\hat{y}_i$ i.e :

$$\epsilon_i = y_i - \hat{y}_i.$$

Propriétés statistiques des estimateurs

1. $E(\hat{\beta}_1) = \beta_1$ d'où $\hat{\beta}_1$ est sans biais.
2. $E(\hat{\beta}_0) = \beta_0$ d'où $\hat{\beta}_0$ est sans biais.
3. $\text{Var}(\hat{\beta}_1) = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$
4. $\text{Var}(\hat{\beta}_0) = \frac{\sigma^2}{n} + \frac{\sigma^2 \bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$.
5. $\mathbb{E}(\hat{\sigma^2}) = \frac{(n-2)}{n}\sigma^2$ d'où σ^2 est biaisé
6. $\text{Cov}(\hat{\beta}_1, \hat{\beta}_0) = \frac{-\bar{x}\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$.

En effet

- **Espérance de $\hat{\beta}_1$:**

$$\begin{aligned}
\mathbb{E}(\hat{\beta}_1) &= \mathbb{E}\left[\frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right] \\
&= \frac{1}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\mathbb{E}\left[\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})\right] \\
&= \frac{1}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\mathbb{E}\left[\sum_{i=1}^{n}(x_i - \bar{x})(\beta_0 + \beta_1 x_i - \beta_0 - \beta_1 \bar{x})\right] \\
&= \frac{1}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\mathbb{E}\left[\sum_{i=1}^{n}(x_i - \bar{x})(\beta_1(x_i - \bar{x}))\right] \\
&= \frac{1}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\mathbb{E}\left[\beta_1 \sum_{i=1}^{n}(x_i - \bar{x})^2\right] \\
&= \frac{1}{\sum_{i=1}^{n}(x_i - \bar{x})^2}(\beta_1 \sum_{i=1}^{n}(x_i - \bar{x})^2) \\
&= \beta_1.
\end{aligned}$$

Donc, $\hat{\beta}_1$ est sans biais.

- **Variance de $\hat{\beta}_1$:**

$$\begin{aligned}
Var(\hat{\beta}_1) &= Var\left(\frac{\sum_{i=1}^n (x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^n (x_i-\bar{x})^2}\right) \\
&= \frac{\sum_{i=1}^n (x_i-\bar{x})^2\, Var(y_i-\bar{y})}{(\sum_{i=1}^n (x_i-\bar{x})^2)^2} \\
&= \frac{1}{\sum_{i=1}^n (x_i-\bar{x})^2}\, Var(y_i-\bar{y}) \\
&= \frac{1}{\sum_{i=1}^n (x_i-\bar{x})^2}\, Var\left(\beta_0+\beta_1 x_i+\epsilon_i-\frac{1}{n}\sum_{i=1}^n \beta_0-\frac{1}{n}\sum_{i=1}^n \beta_1 x_i-\frac{1}{n}\sum_{i=1}^n -\epsilon_i\right) \\
&= \frac{1}{\sum_{i=1}^n (x_i-\bar{x})^2}\, Var\left(\beta_1(x_i-\bar{x})+(\epsilon_i-\bar{\epsilon})\right) \\
&= \frac{1}{\sum_{i=1}^n (x_i-\bar{x})^2}\, Var\left(\beta_1(x_i-\bar{x})+\epsilon_i\right) \\
&= \frac{1}{\sum_{i=1}^n (x_i-\bar{x})^2}\, Var(\epsilon_i) \\
&= \frac{\sigma^2}{\sum_{i=1}^n (x_i-\bar{x})^2}.
\end{aligned}$$

- **Espérance de $\hat{\beta}_0$:**

$$\begin{aligned}
\mathbb{E}(\hat{\beta}_0) &= \mathbb{E}(\overline{y}-\hat{\beta}_1\overline{x}) \\
&= \bar{y}-\mathbb{E}(\hat{\beta}_1)\overline{x} \\
&= (\beta_0+\beta_1\bar{x})-\beta_1\bar{x} \\
&= \beta_0.
\end{aligned}$$

  Donc, $\hat{\beta}_0$ est sans biais.

- **Variance de $\hat{\beta}_0$:**

$$\begin{aligned}
Var(\hat{\beta}_0) &= Var((\overline{y}-\hat{\beta}_1\overline{x}) \\
&= Var(\bar{y})+\bar{x}^2\, Var(\hat{\beta}_1)-2\bar{x}\, Cov(\bar{y},\hat{\beta}_1) \\
&= Var(\bar{y})+\bar{x}^2\, Var(\hat{\beta}_1) \\
&= \frac{\sigma^2}{n}+\frac{\bar{x}^2\sigma^2}{\sum_{i=1}^n (x_i-\bar{x})^2}.
\end{aligned}$$

- **Covariance entre $\hat{\beta}_0$ et $\hat{\beta}_1$:** On a :

$$\begin{aligned}
Cov(\hat{\beta}_1,\hat{\beta}_0) &= Cov(\hat{\beta}_1,\bar{y}-\hat{\beta}_1\bar{x}) \\
&= Cov(\hat{\beta}_1,\bar{y})-\bar{x}\, Var(\hat{\beta}_1) \\
&= 0-\frac{\bar{x}\sigma^2}{\sum_{i=1}^n (x_i-\bar{x})^2} \\
&= \frac{-\bar{x}\sigma^2}{\sum_{i=1}^n (x_i-\bar{x})^2}.
\end{aligned}$$

- **Espérance de $\hat{\sigma^2}$:**

$$\begin{aligned}
\mathbb{E}(\hat{\sigma^2}) &= \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n}[y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)]^2\right) \\
&= \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n}\hat{\epsilon_i}^2\right) = \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right) \\
&= \frac{1}{n}\left(\sum_{i=1}^{n}\mathbb{E}\left((y_i - \hat{y}_i)^2\right)\right) \\
&= \frac{1}{n}\left(\sum_{i=1}^{n}\mathbb{E}(y_i^2) - \sum_{i=1}^{n}\mathbb{E}(\hat{y_i}^2)\right) \\
&= \frac{1}{n}\left(\sum_{i=1}^{n}(\,Var(y_i) + (\mathbb{E}(y_i))^2) - \sum_{i=1}^{n}(\,Var(\hat{y}_i) - (\mathbb{E}(\hat{y}_i))^2)\right).
\end{aligned}$$

Or, on a :

$$\begin{aligned}
\mathbb{E}(\hat{y}_i) &= \mathbb{E}\left(\hat{\beta}_0 + \hat{\beta}_1 x_i\right) \\
&= \mathbb{E}(\hat{\beta}_0) + x_i E(\hat{\beta}_1) \\
&= \beta_0 + x_i\beta_1 \\
&= \mathbb{E}(y_i).
\end{aligned}$$

D'autre part on a :

$$\begin{aligned}
Var(\hat{y}_i) =&\ Var(\hat{\beta}_0 + \hat{\beta}_1 x_i) \\
=&\ Var(\hat{\beta}_0) + x_i^2\, Var(\hat{\beta}_1) + 2x_i\, Cov(\hat{\beta}_1, \hat{\beta}_0) \\
=&\ \frac{\sigma^2}{n} + \frac{\bar{x}^2\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} + \frac{x_i^2\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} + \frac{-2x_i\bar{x}\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \\
=&\ \frac{\sigma^2}{n} + \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\left(\bar{x}^2 + x_i^2 - 2x_i\bar{x}\right) \\
=&\ \frac{\sigma^2}{n} + \frac{\sigma^2(x_i - \bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \\
=&\ \sigma^2\left(\frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right).
\end{aligned}$$

On obtient ainsi :

$$\begin{aligned}
\mathbb{E}(\hat{\sigma^2}) &= \frac{1}{n}\left(\sum_{i=1}^{n}\,Var(y_i) - \sum_{i=1}^{n}\,Var(\hat{y}_i)\right) \\
&= \frac{1}{n}\left(n\sigma^2 - \sigma^2\left(\frac{n}{n} + \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)\right) \\
&= \frac{1}{n}(n\sigma^2 - 2\sigma^2) \\
&= \frac{(n-2)}{n}\sigma^2 \text{ est un estimateur biaisé pour } \sigma^2
\end{aligned}$$

Remarque : Comme $\hat{\sigma^2}$ est un estimateur biaisé pour σ^2, on préfère un autre estimateur appelé "*variance résiduelle*", noté S^2. Il est défini comme suit :

$$S^2 = \frac{1}{n-2}\sum_{i=1}^{n} \hat{\epsilon_i}^2 = \frac{1}{n-2}\sum_{i=1}^{n}\left[Y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)\right]^2,$$

S^2 est un estimateur sans biais pour σ^2, sa variance est : $Var(S^2) = \frac{2\sigma^4}{n-2}$.

Test de signification global de Fisher

Dans ce paragraphe, nous allons tester l'hypothèse

$$H_0 : \beta_1 = 0,$$

contre l'hypothèse alternative :

$$H_1 : \beta_1 \neq 0.$$

En utilisant, trois sommes des carrées SCT, SCE et SCR telles que :

$SCT = \sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2 + \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ est appelé "*la somme des carrés totale*".

$SCE = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$ est appelé "*la somme des carrés expliqués*".

$SCR = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ est appelé "*la somme des carrés des résidus*".

Si cette hypothèse est vérifiée, on peut montrer que :

$$\mathbb{E}(SCT) = (n-1)\sigma^2,$$

$$\mathbb{E}(SCE) = \sigma^2,$$

$$\mathbb{E}(SCR) = (n-2)\sigma^2.$$

On note les estimateurs sans biais de σ^2 par :

$$CMT = \frac{SCT}{n-1},$$

$$CME = \frac{SCE}{1},$$

$$CMR = \frac{SCR}{n-2}.$$

Où CM symbolise la moyenne des carrées. Le nombres $(n-1)$, 1 et $(n-2)$ représentent les degrés de liberté associés à ces sommes de carrées(i.e le nombre de termes linéairement indépendants impliqués dans chacune de ces sommes) et ils représentent aussi le nombre d'éléments dans la somme des carrées moins le nombre de paramètres estimés dans cette somme.
Lorsque H_0 n'est pas vérifiée, la CMR est encore un estimateur sans biais de σ^2, il s'agit en fait de l'estimateur S^2 défini précédemment. Cependant, lorsque H_0 est vérifiée, on a :

$$\frac{SCT}{\sigma^2} \sim \chi^2_{(n-1)},$$

$$\frac{SCE}{\sigma^2} \sim \chi^2_1,$$
$$\frac{SCR}{\sigma^2} \sim \chi^2_{(n-2)}.$$

De plus, les quantités $\frac{SCE}{\sigma^2}$ et $\frac{SCR}{\sigma^2}$ sont indépendantes ainsi, la statistique :

$$F_c = \frac{\left(\frac{SCE}{\sigma^2}\right)}{\left(\frac{SCR}{(n-2)\sigma^2}\right)} = \frac{SCE}{SCR}.\frac{(n-2)}{1} = \frac{CME}{CMR} \sim F_{(1,n-2)}.$$

Par consequent, on rejette H_0 au seuil α si :

$$F_c > F_{(\alpha,1,n-2)},$$

où $F_{(\alpha,1,n-2)}$ est la valeur critique de quantile $(1-\alpha)$ d'une loi $F_{(1,n-2)}$.

Cette procedure est appelée *"analyse de variation"* qui est abrégée par le tableau de l'ANOVA qui se présente dans le tableau 1.1

Source de la variance	Degré de liberté	Somme des carrés	Moyenne des carrés
Régression	1	$SCE = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$	$CME = \frac{SCE}{1}$
Résiduelle	$n-2$	$SCR = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	$CMR = \frac{SCR}{n-2}$
Total	$n-1$	$SCT = \sum_{i=1}^{n}(y_i - \bar{y})^2$	

TABLE 1.1 – Tableau de l'ANOVA pour la régression simple

Coefficient de corrélation

La relation entre deux variables aléatoires X et Y est exprimée par le "*coefficient de corrélation*" ρ" tel que :

$$\rho = \frac{Cov(X,Y)}{\sqrt{Var(X)\,Var(Y)}}.$$

Le coefficient de corrélation d'échantillon est défini par l'expression :

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

Interprétation du coefficient de corrélation

- Si $r = -1$(respectivement $r = 1$) alors, il existe une relation linéaire négative(respectivement positive) entre les deux variables.
- Si $r = 0$ alors, les variables ne sont pas linéairement liées. Cela ne signifie pas nécessairement qu'il n'y a aucune relation entre les variables, mais simplement qu'elle n'est pas linéaire.

- Si $-1 < r < 0$ (respectivement $0 < r < 1$) alors, il existe une corrélation négative (respectivement positive). Les variables sont liées, mais la relation n'est pas parfaitement linéaire.

Remarques :

1. Si ρ est proche de 1 ou de -1 la régression linéaire est justifiée.
2. Dans le cas de la régression linéaire simple, le carré du coefficient de correlation ρ^2 est appelé le "*coefficient de détermination*" noté R^2. Il mesure la proportion de la variance de la variable Y expliquée par le modèle de régression, tel que :

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

1.1.2 Exemple d'application sous langage R

Exemple 1.1.1. Dans cet exemple, on effectue une régression linéaire simple en utilisant des données réelles (112 observations, Rennes, été 2001) où la variable indépendante est la température à 12h du jour et la variable dépendante est la concentration en ozone du lendemain.

Variables observées :

- T9,T12 et T15 → Températures observées à 9, 12 et 15h.
- Ne9, Ne12, Ne15 → Nébulosité observée à 9, 12 et 15h.
- Vx9, Vx12,Vx15 → Composante Est-Ouest du vent à 9, 12 et 15h.
- MaxO3v → Teneur maximum en ozone observée la veille.
- vent → orientation du vent à 12h.
- Pluie → occurrence ou non de précipitations.
- MaxO3 → concentration max.d'ozone observée sur la journée.

```r
# Load the data
ozone <- read.table("C:/Program␣Files/R/R-4.4.0/bin/x64/ozone.txt",
    header = TRUE)
```

```r
summary(ozone[, c("maxO3", "T12")])
```

```
     maxO3            T12
 Min.   : 42.00   Min.   :14.00
 1st Qu.: 70.25   1st Qu.:18.60
 Median : 81.00   Median :20.65
 Mean   : 90.35   Mean   :21.57
 3rd Qu.:106.00   3rd Qu.:23.65
 Max.   :166.00   Max.   :33.50
```

Interprétation : Ces statistiques nous donnent une idée de la distribution des valeurs pour chaque variable. Pour maxO3, les valeurs se situent entre 42.00 et 166.00, avec une médiane de 81.00 et une moyenne de 90.35 et pour T12, les valeurs varient entre 14.00 et 33.50, avec une médiane de 20.65 et une moyenne de 21.57.

- **Diagramme de dispersion de maxO3 en fonction de T12 :**

```
plot(maxO3 ~ T12, data = ozone, pch = 15, cex = 0.5)
```

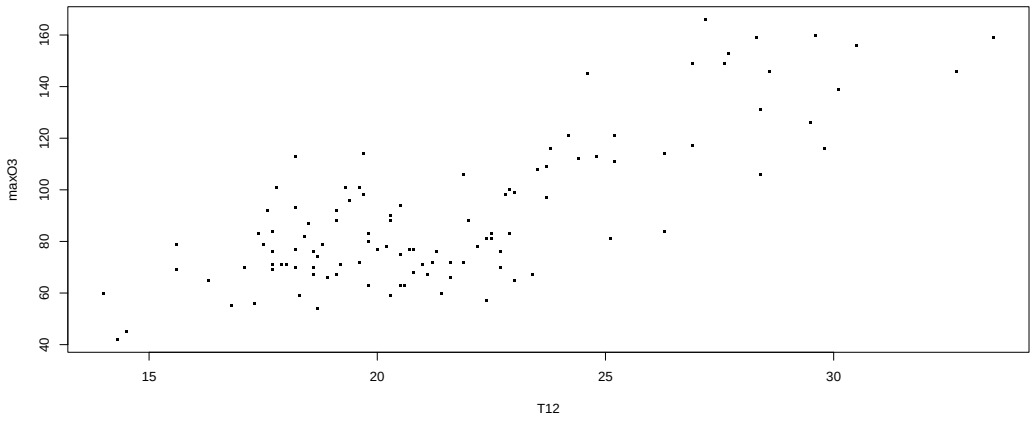

FIGURE 1.1 – Représentation graphique de nuage de points.

- **Ajout d'une courbe lissée au diagramme de dispersion :**

```
scatter.smooth(x = ozone$T12, y = ozone$maxO3)
```

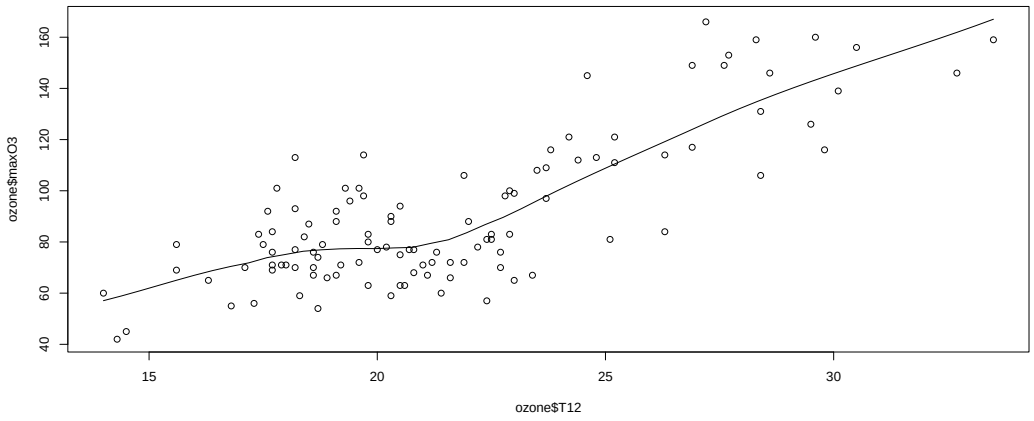

FIGURE 1.2 – Courbe de lissage .

Ce graphique représente une visualisation relative à la température et aux niveaux maximaux d'ozone.

- **Ajustement d'un modèle de régression linéaire simple :**

```
reg.simple <- lm(maxO3 ~ T12, data = ozone)
summary(reg.simple)
```

```
Call:
lm(formula = maxO3 ~ T12, data = ozone)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-37.912 -12.746   0.504  11.417  44.593

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -28.7321     9.1056  -3.155  0.00208 **
T12           5.5198     0.4149  13.305  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 17.59 on 108 degrees of freedom
Multiple R-squared:  0.6211,    Adjusted R-squared:  0.6176
F-statistic:   177 on 1 and 108 DF,  p-value: < 2.2e-16
```

Interprétation des résultats : Nous remarquons que les résidus vont de -37.912 à 44.593 avec une distribution des résidus au tour de la médiane 0.504. Les coefficients montrent que l'estimation de l'interception est -28.7321, ce qui signifie que lorsque T12 est nulle la valeur de maxO3 est -28.7321 ce qui peut ne pas être pertinente dans un contexte réel.

Le coefficient de T12 est 5.5198 indique qu'une augmentation d'une unité de T12 correspond à une augmentation de 5.5198 unités de maxO3. Nous observons aussi que toutes les p-values des erreurs standard et des valeurs t sont très faibles (< 0.05), cela indique qu'il existe une forte signification statistique des coefficients. L'erreur standard des résidus est de 17.59 indique la dispersion des résidus au tour de la ligne de régression.

Le R^2 de 0.6211 montre que 62.11% de la variabilité de maxO3 est expliquée par T12 et la statistique F de p-value inférieur à $2e-16$ indique que le modèle est globalement significatif.

En résumé, le modèle révèle une relation positive et significative entre T12 et maxO3 avec une proportion importante de la variabilité de maxO3 expliquée par T12.

- **Extraction des coefficients du modèle :**

```
1 reg.simple$coefficients
```

```
(Intercept)         T12
 -28.732149    5.519821
```

Interprétation : Le modèle final est donné par :

$$maxO3 = 5.519821 * T12 - 28.7321.$$

- **Génération d'une séquence de valeurs pour T12 et Calcul des valeurs prédites de maxO3 :**

```
x <- seq(min(ozone$T12), max(ozone$T12), length = 100)
y <- reg.simple$coef[1] + reg.simple$coef[2] * x
# Add the regression line to the scatter plot
lines(x, y, col = 2)
```

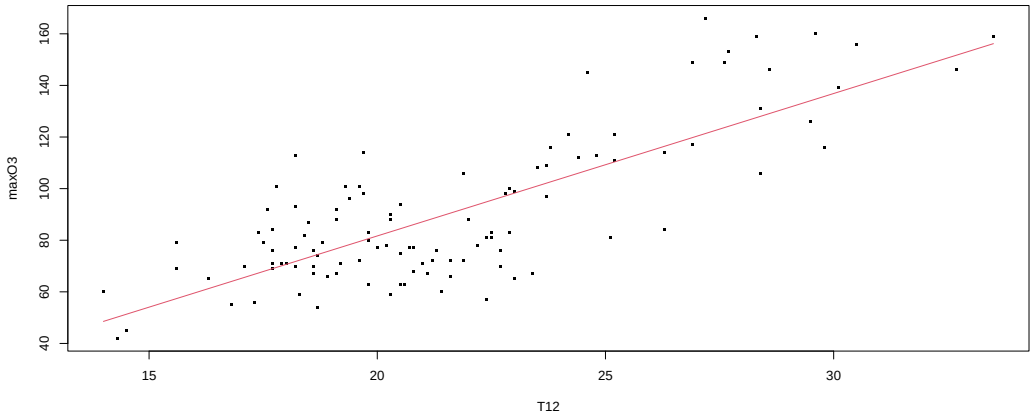

FIGURE 1.3 – Représentation graphique de la régression linéaire .

Interprétation : Ce graphique confirme les résultats de l'erreur standard des résidus.

- **Analyse de la variance du modèle de régression linéaire :**

```
anova(reg.simple)
```

```
Analysis of Variance Table

Response: maxO3
           Df Sum Sq Mean Sq F value    Pr(>F)
T12         1  54747   54747  177.02 < 2.2e-16 ***
Residuals 108  33401     309
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Interprétation : Ces résultats confirment que la variation de la variable maxO3 est significativement expliquée par la variable T12 dans le modèle de régression linéaire.

1.1.3 Régression linéaire multiple

Modèle

Dans le contexte de la régression linéaire multiple, on cherche à étudier la relation entre la variable quantitative Y et p variables quantitatives $X^1, ..., X^p$.
Les données sont issues de l'observation d'un échantillon statistique de taille $n(n > p+1)$ dans l'espace $\mathbb{R}^{p+1}$ i,e :

$$(x_i^1, x_i^2, ..., x_i^j, ..., x_i^p, y_i) \quad \forall i = 1, ..., n.$$

Le modèle dans cette situation est défini sous la forme :

$$Y_i = \beta_0 + \beta_1 x_i^1 + \beta_2 x_i^2 + ... + \beta_{p-1} x_i^{p-1} + \epsilon_i \quad \forall i = 1, .., n \tag{1.3}$$

où :

1. Les ϵ_i sont les erreurs, $i.i.d$ de loi $N(0, \sigma_\epsilon^2)$.
2. Les termes X^j sont déterminés, alternativement, on suppose que l'erreur ϵ indépendante de la distribution conjointe de $X^1, ..., X^p$. Cela signifie que :

$$\mathbb{E}(Y|X^1, ..., X^p) = \beta_0 + \beta_1 X^1 + ... + \beta_p X^p \quad \text{et} \quad Var(Y|X^1, ..., X^p) = \sigma_\epsilon^2.$$

On peut réécrire le modèle (1.3) sous la forme matricielle afin de simplifier la résolution du problème. Cette formulation est définie par :

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{p-1} \\ 1 & x_{21} & x_{22} & \dots & x_{2p-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np-1} \end{bmatrix} \times \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Estimation des paramètres du modèle de régression

Le but est d'estimer le vecteur β par un vecteur d'estimateurs $\hat{\beta}$, tel que :

$$\hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_{p-1} \end{bmatrix}$$

soit $\hat{Y}$ le vecteur des valeurs estimées tel que :

$$\hat{Y} = \begin{bmatrix} \hat{y_1} \\ \hat{y_2} \\ \vdots \\ \hat{y_n} \end{bmatrix}$$

ainsi, en définissant

$$\hat{Y} = X\hat{\beta}$$

et $\hat{\epsilon}$ est le vecteur des résidus tel que :

$$\hat{\epsilon} = \begin{bmatrix} \hat{\epsilon_1} \\ \hat{\epsilon_2} \\ \vdots \\ \hat{\epsilon_n} \end{bmatrix}$$

en posant :

$$\hat{\epsilon} = Y - \hat{Y}.$$

La régression linéaire multiple consiste à résoudre :

$$\hat{\beta} = Arg\min_{\beta} \sum_{i=1}^{n} \hat{\epsilon_i}^2 \quad \text{(critère des moindres carrés)}.$$

Supposons que $Q(\beta) = \sum_{i=1}^{n} \epsilon_i^2$ tel que :

$$\begin{aligned} Q(\beta) = \sum_{i=1}^{n} \epsilon_i^2 = \epsilon^{'}\epsilon &= (Y - X\beta)^{'}(Y - X\beta) \\ &= Y^{'}Y - \beta^{'}X'Y - Y^{'}X\beta + \beta^{'}X^{'}X\beta \\ &= Y^{'}Y - 2\beta^{'}X^{'}Y + \beta^{'}X^{'}X\beta. \end{aligned}$$

Comme $\beta^{'}X^{'}Y$ est un nombre, ce qui implique que $\beta^{'}X^{'}Y = Y^{'}X\beta$.

Or :

$$\hat{\beta} = Arg\min_{\beta} \sum_{i=1}^{n} \hat{\epsilon_i}^2.$$

Par conséquent, nous devons calculer la dérivée partielle de la fonction $Q(\beta)$ par rapport à β, qui est donnée par :

$$-2X^{'}Y + 2X^{'}X\beta.$$

On a :

$$\frac{\partial Q(\beta)}{\partial \beta} = 0 \Leftrightarrow \hat{\beta} = (X^{'}X)^{-1}X^{'}Y$$

et on a aussi :

$$\frac{\partial^2 Q(\beta)}{\partial \beta \partial \beta'} = 2X^{'}X \quad \text{qui est définie positive.}$$

Donc, $\hat{\beta} = (X^{'}X)^{-1}X^{'}Y$ est bien un estimateur de maximum de vraisemblance et également est un estimateur des moindres carrés car les ϵ_i sont gaussiens.

Finalement,

$$\hat{Y} = X\hat{\beta} = X(X^{'}X)^{-1}X^{'}Y = P_X Y$$

où P_X est une matrice idempotente.

Rappels :

1. **Dérivation matricielle :**
   (a) Si a un vecteur constant, alors :
   $$\frac{\partial(a^{'}u)}{\partial u} = a.$$
   (b) Soit A une matrice de $(p * p)$, alors :
   $$\frac{\partial(u^{'}Au)}{\partial u} = Au + A^{'}u.$$
   (c) Si $A = A^{'}$ (symétrique), alors :
   $$\frac{\partial(u^{'}Au)}{\partial u} = 2Au.$$
2. **Matrice idempotente :**
   (a) On dit que A est une matrice idempotente, si $A^2 = A$.

Propriétés statistiques de $\hat{\beta}$

- **Espérance de $\hat{\beta}$:**

$$\begin{aligned}\mathbb{E}(\hat{\beta}) &= \mathbb{E}((X'X)^{-1}X^{'}Y) \\ &= \mathbb{E}((X'X)^{-1}X^{'}(X\beta + \epsilon)) \\ &= \mathbb{E}((X'X)^{-1}X^{'}X\beta + (X'X)^{-1}X^{'}\epsilon) \\ &= \mathbb{E}(\beta) \\ &= \beta\end{aligned}$$

$$\Rightarrow \hat{\beta} = (X'X)^{-1}X^{'}Y \text{ est un estimateur sans biais de } \beta.$$

- **Variance de $\hat{\beta}$:**

$$\begin{aligned}Var(\hat{\beta}) &= Var((X'X)^{-1}X^{'}Y) \\ &= Var((X'X)^{-1}X^{'}(X\beta + \epsilon)) \\ &= Var((X'X)^{-1}X^{'}X\beta + (X'X)^{-1}X^{'}\epsilon) \\ &= Var(\beta) + Var((X'X)^{-1}X^{'}\epsilon) \\ &= (X'X)^{-1}X^{'}\,Var(\epsilon)X(X^{'}X)^{-1} \\ &= \sigma^2(X'X)^{-1}.\end{aligned}$$

Rappel : Soit A une matrice de coefficients et u un vecteur aléatoire alors :

$$Var(Au) = A\,Var(u)A'.$$

Théorème 1. ***[5]***
Parmi tous les estimateurs linéaires en Y sans biais de β, $\hat{\beta}$ est celui qui a la variance minimale.

En effet, soit $\tilde{\beta}$ est un estimateur sans biais de β avec $\tilde{\beta} = AY$ où $A = (X'X)^{-1}X'$ et $\mathbb{E}(\tilde{\beta}) = \beta$

$$\begin{aligned} Var(\tilde{\beta}) &= Var(\tilde{\beta} - \hat{\beta} + \hat{\beta}) \\ &= Var(\tilde{\beta} - \hat{\beta}) + Var(\hat{\beta}) + Cov(\tilde{\beta} - \hat{\beta}, \hat{\beta}) + Cov(\hat{\beta}, \tilde{\beta} - \hat{\beta}). \end{aligned}$$

Or :

$$\begin{aligned} Cov(\tilde{\beta} - \hat{\beta}, \hat{\beta}) &= \mathbb{E}((\tilde{\beta} - \hat{\beta})\hat{\beta}') - \mathbb{E}(\tilde{\beta} - \hat{\beta})\mathbb{E}(\hat{\beta}') \\ &= \mathbb{E}(\tilde{\beta}\hat{\beta}') - \mathbb{E}(\hat{\beta}\hat{\beta}') - \Big(\mathbb{E}(\tilde{\beta}) - \mathbb{E}(\hat{\beta})\Big)\mathbb{E}(\hat{\beta}') \\ &= \mathbb{E}(\tilde{\beta}\hat{\beta}') - Var(\hat{\beta}) - (\beta - \beta)\mathbb{E}(\hat{\beta}') \\ &= \mathbb{E}(\tilde{\beta}\hat{\beta}') - Var(\hat{\beta}) \\ &= \mathbb{E}\left(AYY'X(X'X)^{-1}\right) - \sigma^2(X'X)^{-1} \\ &= A\mathbb{E}(YY')X(X'X)^{-1} - \sigma^2(X'X)^{-1} \\ &= \sigma^2(X'X)^{-1}X'X(X'X)^{-1} - \sigma^2(X'X)^{-1} \\ &= \sigma^2(X'X)^{-1} - \sigma^2(X'X)^{-1} \\ &= 0. \end{aligned}$$

De la même manière,

$$Cov(\hat{\beta}, \tilde{\beta} - \hat{\beta}) = 0.$$

$$\text{Donc, } Var(\tilde{\beta}) = Var(\tilde{\beta} - \hat{\beta}) + Var(\hat{\beta}) \geq Var(\hat{\beta}).$$

Remarque : L'estimateur sans biais de σ^2 est donné par :

$$\hat{\sigma^2} = \frac{1}{n-p}\hat{\epsilon}'\hat{\epsilon}.$$

Analyse de variation (ANOVA)

En considérant un modèle de type (1.3), on peut montrer que si l'hypothèse

$$H_0 = \beta_1 = \beta_2 = ... = \beta_{p-1}$$

est vérifiée, alors on peut montrer que les espérances des trois sommes des carrés définies précédemment sont respectivement :

$$\mathbb{E}(SCT) = (n-1)\sigma^2,$$

$$\mathbb{E}(SCE) = (p-1)\sigma^2,$$

$$\mathbb{E}(SCR) = (n-p)\sigma^2.$$

Et les estimateurs sans biais de la variance des erreurs σ^2 sont définies comme suit :

$$CMT = \frac{SCT}{n-1},$$

$$CME = \frac{SCE}{p-1},$$

$$CMR = \frac{SCR}{n-p-1}.$$

Lorsque H_0 n'est pas vérifiée, le seul estimateur sans biais de σ^2 est la CMR, mais lorsque H_0 est vérifiée, on peut montrer de la même façon que dans le cas de la régression simple, que la statistique :

$$F_c = \frac{CME}{CMR} \sim F_{(p-1,n-p)}.$$

Ainsi, on rejette H_0 à un seuil α si :

$$F_c > F_{(\alpha,p-1,n-p)},$$

où la valeur critique $F_{(\alpha,p-1,n-p)}$ est le $(1-\alpha)$ quantile d'une loi de Fisher avec deux degrés de liberté $(p-1)$ et $(n-p)$ que l'on trouve dans une table de Fisher.
Le tableau ci-dessous représente le tableau de l'analyse de variation(ANOVA) lors d'une régression multiple.

Source de la variance	Degré de liberté	Somme des carrés	Moyenne des carrés
Régression	$p-1$	$SCE = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$	$CME = \frac{SCE}{p-1}$
Résiduelle	$n-p$	$SCR = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	$CMR = \frac{SCR}{n-p}$
Total	$n-1$	$SCT = \sum_{i=1}^{n}(y_i - \bar{y})^2$	

TABLE 1.2 – Tableau de l'ANOVA pour la régression multiple.

Remarque : Si on pose $p = 2$, on retrouve le tableau de l'ANOVA de la régression simple.

Coefficient de détermination

Rappelons que le coefficient de détermination R^2 est défini comme suit :

$$R^2 = \frac{SCE}{SCT}$$

Il mesure la proportion de la variance de la variable dépendante Y expliquée par le modèle de régression. Géométriquement, il exprime le rapport des carrés des normes de deux vecteurs : le vecteur Y et sa projection $\hat{Y}$ sur l'espace engendré par les variables indépendantes X. Lorsque R^2 est proche de 1, cela signifie que le modèle explique une grande partie de la variabilité de Y et donc nous conservons plus d'informations.
Cependant, le coefficient R^2 est biaisé, ce qui justifie l'introduction du coefficient de détermination ajusté R^2_{adj} tel que :

$$R^2_{adj} = 1 - \left(\frac{(1-R^2)(n-1)}{n-p-1}\right)$$

Remarque : Dans cette partie, nous allons utiliser le cas où les ϵ_i sont gaussiens mais, nous pouvons suivre la même procédure quelle que soit la loi des ϵ_i.

1.1.4 Exemple sous langage R

Exemple 1.1.2. Nous allons effectuer une régression linéaire multiple sur les données de l'exemple précédent (exemple 1.1.1). Dans la première approche, nous utilisons toutes les variables observées et dans la deuxième, nous sélectionnons les variables.

- **Première approche avec toutes les variables :**

```
#Multiple linear regression with all variables
reg.mul<-lm(maxO3~.,data=ozone)
summary(reg.mul)
```

```
lm(formula = maxO3 ~ ., data = ozone)

Residuals:
    Min      1Q  Median      3Q     Max
-51.754  -8.422  -0.983   8.061  40.179

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 15.121881  16.883904   0.896   0.3727
T9           0.050394   1.176331   0.043   0.9659
T12          2.001547   1.503368   1.331   0.1863
T15          0.439872   1.200346   0.366   0.7148
Ne9         -2.061846   0.990802  -2.081   0.0401 *
Ne12        -0.661129   1.444912  -0.458   0.6483
Ne15         0.001136   1.048665   0.001   0.9991
Vx9          0.435057   1.003905   0.433   0.6657
Vx12         0.607970   1.276977   0.476   0.6351
Vx15         0.732983   0.974292   0.752   0.4537
maxO3v       0.340524   0.068154   4.996 2.66e-06 ***
ventNord     1.420588   7.170042   0.198   0.8434
ventOuest    6.625860   8.752160   0.757   0.4509
ventSud      6.507981   7.635006   0.852   0.3961
pluieSec     3.264895   3.532254   0.924   0.3577
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.64 on 95 degrees of freedom
Multiple R-squared:  0.769,     Adjusted R-squared:  0.7349
F-statistic: 22.58 on 14 and 95 DF,  p-value: < 2.2e-16
```

Interprétation des résultats : Nous observons que la variable Ne9 est statistiquement significative de p-value 0.0401, ce qui suggère une relation négative avec maxO3. De plus, la variable maxO3v a un impact positif sur la variable maxO3 de p-value $2.2e-16$ hors les autres variables n'ont pas une significativité statistique importante.

Le coefficient de détermination multiple indique que 76.9% de la variance de maxO3 est expliquée par ce modèle, tandis que l'erreur standard des résidus est de 14.64 ce qui mesure la dispersion des valeurs observées autour des valeurs prédites.

Le F-statistic de 22.58 de p-value inférieur à 0.05 indique que le modèle global est statistiquement significatif.

```
anova(reg.mul)
```

```
Analysis of Variance Table

Response: maxO3
          Df Sum Sq Mean Sq  F value    Pr(>F)
T9         1  43545   43545 203.1223 < 2.2e-16 ***
T12        1  11227   11227  52.3696 1.178e-10 ***
T15        1    785     785   3.6631 0.0586414 .
Ne9        1   2932    2932  13.6781 0.0003632 ***
Ne12       1    262     262   1.2217 0.2718247
Ne15       1      1       1   0.0070 0.9335163
Vx9        1   2118    2118   9.8808 0.0022285 **
Vx12       1      0       0   0.0019 0.9653612
Vx15       1     56      56   0.2616 0.6102252
maxO3v     1   6411    6411  29.9034 3.662e-07 ***
vent       3    261      87   0.4064 0.7487559
pluie      1    183     183   0.8543 0.3576671
Residuals 95  20366     214
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interprétation des résultats : Ces résultats montrent que les variables T9, T12, Ne9, Vx9 et maxO3 ont un impact important sur la variable maxO3 tandis que les autres variables ne montrent pas de significativité statistique dans ce modèle.

Le faible F value et p-value des résidus indiquent que les variations de maxO3 ne sont pas bien expliquées par ces variables résiduelles.

- **Deuxième approche avec les variables sélectionnées :**

```
#Linear regression with selected variables only
reg.fin<-lm(maxO3~T12+Ne9+Vx9+maxO3v,data=ozone)
summary(reg.fin)
```

```
Call:
lm(formula = maxO3 ~ T12 + Ne9 + Vx9 + maxO3v, data = ozone)

Residuals:
    Min      1Q  Median      3Q     Max
-52.372  -8.390  -1.326   8.011  40.913
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  12.1648    11.5173   1.056 0.293290
T12           2.7830     0.4895   5.685 1.18e-07 ***
Ne9          -2.4999     0.6915  -3.615 0.000463 ***
Vx9           1.2756     0.6155   2.072 0.040673 *
maxO3v        0.3542     0.0586   6.045 2.31e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.13 on 105 degrees of freedom
Multiple R-squared:  0.7622,    Adjusted R-squared:  0.7532
F-statistic: 84.15 on 4 and 105 DF,  p-value: < 2.2e-16
```

Interprétation des résultats : Les résultats de ce modèle indiquent que toutes les variables ont un effets significatif sur le niveau de maxO3. Les variables T12,Vx9 et maxO3 ont des effets positifs significatifs tandis que Ne9 a un impact négatif significatif.

Le R^2montre que 76.22% de la variance de maxO3 est expliquée par ce modèle. L'erreur standard des résidus est de 14.13 indiquant une dispersion modeste des valeurs observées autour des valeurs prédites et le F-statistic de 84.15 de p-value inférieur à $2.2e-16$ montre que le modèle est bien ajusté aux données.

conclusion : Le modèle final est donné par :

$maxO3 = 12.1648 + 2.7830 * T12 - 2.4999 * Ne9 + 1.2756 * Vx9 + 0.3542 * maxO3v.$

1.2 Régression polynomiale

La régression polynomiale est une méthode statistique qui permet de capturer des relations non linéaires entre la *variable expliquée* Yet une ou plusieurs *variables explicatives* X.
Généralement, dans la régression polynomiale, on ajuste un polynôme de degré p à partir des données. Ce polynôme prend la forme suivante :

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + ... + \beta_p x_i^p + \epsilon_i \tag{1.4}$$

Le but de cette régression est de trouver les valeurs des coefficients $\beta_0, \beta_1, ..., \beta_p$ qui minimisent l'erreur entre les valeurs prédites par le modèle et les valeurs observées dans les données réelles.

Dans cette section, nous traiterons le cas d'un polynôme de degré $p = 2$.

1.2.1 Présentation du modèle

Le modèle de la régression polynomiale de degré $p = 2$ s'écrit sous la forme suivante :

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i \tag{1.5}$$

où :

- Les β_j pour tout $j = 0, 1, 2$ sont des coefficients inconnues à estimer
- Les x_i^2 représentent le terme quadratique, ce qui permet au modèle de capturer des relations non linéaires entre X et Y
- Les ϵ_i sont les erreurs résiduelles.

On peut réécrire le modèle (1.5) sous la forme matricielle afin de simplifier la résolution du problème. Cette formulation est définie par :

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12}^2 \\ 1 & x_{21} & x_{22}^2 \\ \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2}^2 \end{bmatrix} \times \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

1.2.2 Estimation des paramètres β_0, β_1 et β_2

L'objectif est de trouver les valeurs des coefficients β_0, β_1 et β_2 qui minimisent la somme des carrés des erreurs résiduelles i.e :

$$\hat{\beta} = Arg \min_{\beta} \sum_{i=1}^{n} \left(y_i - (\beta_0 + \beta_1 x_i + \beta_2 x_i^2)\right)^2 .$$

Posons :

$$f(\beta_0, \beta_1, \beta_2) = \sum_{i=1}^{n} \left(y_i - (\beta_0 + \beta_1 x_i + \beta_2 x_i^2)\right)^2 .$$

- **L'estimateur de β_0 est :**
  Appliquons la dérivation par rapport à β_0 :

$$\frac{\partial f(\beta_0, \beta_1, \beta_2)}{\partial \beta_0} = -2 \sum_{i=1}^{n} \left(y_i - (\beta_0 + \beta_1 x_i + \beta_2 x_i^2)\right).$$

  Ensuite, nous égalons cette dérivée à zéro et résolvons pour β_0 :

$$\begin{aligned} \frac{\partial f(\beta_0, \beta_1, \beta_2)}{\partial \beta_0} = 0 &\Leftrightarrow \sum_{i=1}^{n} \left(y_i - (\beta_0 + \beta_1 x_i + \beta_2 x_i^2)\right) = 0 \\ &\Leftrightarrow \sum_{i=1}^{n} y_i - n\beta_0 - \beta_1 \sum_{i=1}^{n} x_i - \beta_2 \sum_{i=1}^{n} x_i^2 = 0 \\ &\Leftrightarrow n\beta_0 = \sum_{i=1}^{n} y_i - \beta_1 \sum_{i=1}^{n} x_i - \beta_2 \sum_{i=1}^{n} x_i^2 \\ &\Rightarrow \hat{\beta}_0 = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - (\beta_1 x_i + \beta_2 x_i^2)\right). \end{aligned}$$

  La dérivée seconde par rapport à β_0 est donnée par :

$$\left.\frac{\partial^2 f(\beta_0, \beta_1, \beta_2)}{\partial \beta_0^2}\right|_{\beta_0 = \hat{\beta}_0} = 2n > 0.$$

$$\text{Donc, } \hat{\beta}_0 = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (\beta_1 x_i + \beta_2 x_i^2)\right) \text{ est un minimum.}$$

Par le même raisonnement, on trouve $\hat{\beta}_1$ et $\hat{\beta}_2$:

- **L'estimateur de β_1 est :**

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i\left(y_i - (\beta_0 + \beta_2 x_i^2)\right)}{\sum_{i=1}^{n} x_i^2}.$$

- **L'estimateur de β_2 est :**

$$\hat{\beta}_2 = \frac{\sum_{i=1}^{n} x_i^2\left(y_i - (\beta_0 + \beta_1 x_i)\right)}{\sum_{i=1}^{n} x_i^4}.$$

On peut représenter ces paramètres sous la forme matricielle :
On sait que :

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

où :

$$X^{'}X = \begin{bmatrix} n & S_x & S_{x^2} \\ S_x & S_{x^2} & S_{x^3} \\ S_{x^2} & S_{x^3} & S_{x^4} \end{bmatrix}$$

et

$$X^{'}Y = \begin{bmatrix} S_y \\ S_{xy} \\ S_{x^2y} \end{bmatrix}$$

avec :

$$S_x = \sum_{i=1}^{n} x_i, \quad S_y = \sum_{i=1}^{n} y_i, \quad S_{x^2} = \sum_{i=1}^{n} x_i^2, \quad S_{xy} = \sum_{i=1}^{n} x_i y_i,$$

$$S_{x^3} = \sum_{i=1}^{n} x_i^3, \quad S_{x^2y} = \sum_{i=1}^{n} x_i^2 y_i, \quad S_{x^4} = \sum_{i=1}^{n} x_i^4.$$

Alors, son écriture matricielle est :

$$\begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix} = \begin{bmatrix} n & S_x & S_{x^2} \\ S_x & S_{x^2} & S_{x^3} \\ S_{x^2} & S_{x^3} & S_{x^4} \end{bmatrix}^{-1} \times \begin{bmatrix} S_y \\ S_{xy} \\ S_{x^2y} \end{bmatrix}$$

Remarque : La régression linéaire est un cas particulier de la regression polynomiale où $p = 1$.

1.2.3 Exemple d'application

Exemple 1.2.1. Dans cet exemple, nous comparons deux modèle de régression : la régression linéaire et la régression polynomiale.

Soit un ensemble de points $(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_{11}, y_{11})$ tel que :

x	0	1	2	3	4	5	6	7	8	9	10
y	10	8	15	15	30	40	44	60	80	85	110

- Ces données sont tracées sur la figure ci-dessous

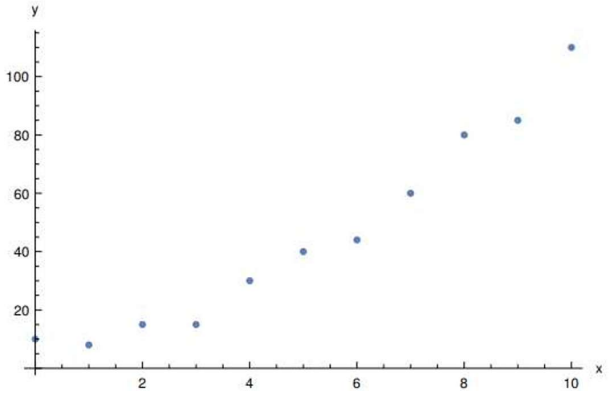

FIGURE 1.4 – Représentation graphique des données.

- **Régression polynomiale :**
  1. Calculons $\hat{\beta}_0, \hat{\beta}_1$ et $\hat{\beta}_2$:

On a :

$$S_x = 55, \quad S_y = 497, \quad S_{x^2} = 385, \quad S_{xy} = 3592,$$
$$S_{x^3} = 3025, \quad S_{x^2y} = 29212, \quad S_{x^4} = 25333,$$

$$\begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix} = \begin{bmatrix} 11 & 55 & 385 \\ 55 & 385 & 3025 \\ 385 & 3025 & 25333 \end{bmatrix}^{-1} \times \begin{bmatrix} 497 \\ 3592 \\ 29212 \end{bmatrix}$$

$$\Rightarrow \begin{cases} \hat{\beta}_0 = \frac{103}{13} \approx 7.93 \\ \hat{\beta}_1 = \frac{1941}{1430} \approx 1.357 \\ \hat{\beta}_2 = \frac{249}{286} \approx 0.871 \end{cases}$$

Donc, $\hat{y} = 7.923 + 1.357x + 0.71x^2$.

Cette parabole est tracée dans la figure (1.5)

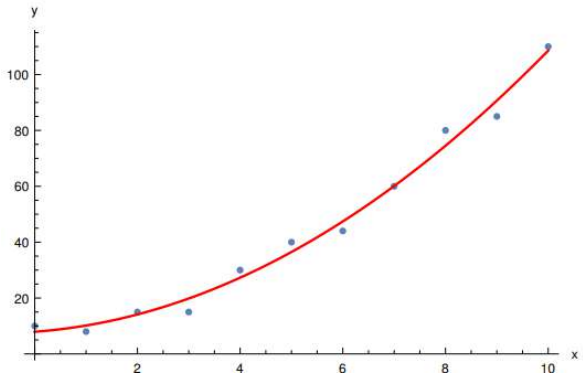

FIGURE 1.5 – Représentation graphique de la régression polynomiale.

2. Calculons le coefficient de détermination R^2 :

$$R^2 = \frac{SCE}{SCT} = 1 - \frac{SCR}{SCT} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

où :

$$\overline{y} = \frac{1}{11}\sum_{i=1}^{11} y_i \approx 45.182$$

$$SCR = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \approx 294.011$$

$$SCT = \sum_{i=1}^{n}(y_i - \bar{y})^2 \approx 11919.6$$

$$\Rightarrow R^2 = 1 - \frac{SCR}{SCT} \approx 0.975$$

- **Régression linéaire :**
  On sait que la régression linéaire est une régression polynomiale de degré $p = 1$:

  1. Trouver les estimateurs des paramètres β_0 et β_1 :

$$\begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix} = \begin{bmatrix} n & S_x \\ S_x & S_{x^2} \end{bmatrix}^{-1} \times \begin{bmatrix} S_y \\ S_{xy} \end{bmatrix} = \begin{bmatrix} 11 & 55 \\ 55 & 385 \end{bmatrix}^{-1} \times \begin{bmatrix} 498 \\ 3592 \end{bmatrix}$$

$$\Rightarrow \begin{cases} \hat{\beta}_0 = \frac{1108}{110} \approx 10.064 \\ \hat{\beta}_1 = \frac{-113}{22} \approx -5.136 \end{cases}$$

Donc, $\hat{y} = 10.064 - 5.136x$

Cette droite est tracée dans la figure (1.6)

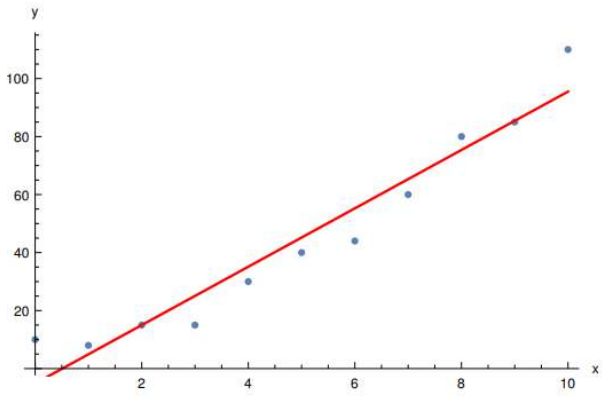

FIGURE 1.6 – Représentation graphique de la régression linéaire.

2. Calculons le coefficient de détermination R^2

$$R^2 = 1 - \frac{SCR}{SCT} \approx 0.935$$

- **Comparaison de ces deux modèles de régression :**
  Nous remarquons que la valeur de R^2 de la régression polynomiale est beaucoup plus proche de 1 que celle obtenue avec la régression linéaire. Cette observation confirme que le modèle polynomial de degré 2 est un bien plus adapté à nos données.

II. Régression non paramétrique

La régression non paramétrique est une approche statistique qui permet d'adapter une courbe ou une fonction à un ensemble de données sans faire d'hypothèses sur la forme fonctionnelle de la relation entre les variables indépendantes et dépendantes. Elle est ainsi un outil polyvalent pour modéliser des relations complexes qui peuvent pas être capturées par de simples modèles linéaires ou polynomiaux.

Il existe plusieurs types de méthodes de régression non paramétriques qui peuvent être utilisées pour modéliser les données parmi elles : régression du noyau, régression locale, régression spline(voir chapitre 2), chacune ayant ses propres forces et faiblesses. Le choix de la méthode dépend de la nature des données et des objectifs de l'analyse. Dans ce paragraphe nous nous concentrons sur la régression du noyau.

1.2.1 Présentation du modèle

Soit n observations $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ de couple (X, Y).
En general, dans le modèle de régression non paramétrique nous supposons l'existence d'une fonction $m(.)$ qui exprime la valeur moyenne de la variable réponse Y en fonction de la variable explicative X. Ce modèle est défini comme suit :

$$y_i = m(x_i) + \epsilon_i \quad \forall i = 1, ..., n. \tag{1.6}$$

où $m(x_i)$ est la fonction à estimer et les ϵ_i sont des variables aléatoires (erreurs) de loi $N(0, \sigma^2)$.

On peut le réécrire sous la forme d'une espérance conditionnelle telle que :

$$E(Y|X = x) = m(x) \tag{1.7}$$

où m est k fois continûment dérivable. k étant un entier positif ou nul (le cas $k = 0$ correspondant évidemment à l'hypothèse de continuité de m).

1.2.2 Modèle de régression non paramétrique(régression à noyau de Nadaraya-Watson)

Principe de la méthode

Le problème consiste à estimer la fonction de régression en tous points $x_1, x_2, ..., x_n$. Le principe de la méthode du noyau repose sur des techniques de lissage, elle donne pour estimateur de $\mathbb{E}(Y|X = x)$ une moyenne pondérée des valeurs y_i pour les i dont le point x_i est proche du point d'estimation. Le choix du point d'estimation x_0, qui représente la valeur spécifique de x pour laquelle nous cherchons à estimer $m(x_0)$, la fonction de régression.

Estimateur de Nadaraya-Watson

Soit un ensemble d'observations $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$. On appelle *estimateur de Nadaraya-Watson* tout estimateur à noyau simple, noté $\hat{m}_{NW}(x)$. Il est défini comme suit :

$$\hat{m}_{NW}(x) = \frac{\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right) y_i}{\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)} \times \mathbb{1}\left\{\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right) \neq 0\right\} \quad h > 0.$$

On rappelle que :

$$\mathbb{1}(A) = \begin{cases} 1, & \text{si } A \text{ est vérifié,} \\ 0, & \text{sinon.} \end{cases}$$

Alors, l'estimateur de Nadaraya-Watson est défini par :

$$\hat{m}_{NW}(x) = \begin{cases} \frac{\sum_{i=1}^{n} y_i K\left(\frac{x-x_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)}, & \text{si } \sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right) \neq 0, \\ \frac{1}{n}\sum_{i=1}^{n} y_i, & \text{sinon.} \end{cases}$$

avec $K : \mathbb{R} \mapsto \mathbb{R}$ est supposée mesurable et satisfaisant certaines hypothèses basiques parmi celles énoncées ci-dessous :

- K est bornée, i.e. $\sup_{u\in\mathbb{R}} |K(u)| < \infty$
- $\lim_{|u|\to\infty} |u|K(u) = 0 \quad$ (K.2)
- $\int_{\mathbb{R}} |K(u)|\, du < \infty$
- $\int_{\mathbb{R}} K(u)\, du = 1$

et h est paramètre de lissage.

Proposition 1 : On appelle estimateur à noyau (kernel estimate) de Nadaraya-Watson tout estimateur sous la forme suivante :

$$\hat{m}(x_0) = \sum_{i=1}^{n} w_i(x_0) y_i$$

avec :

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

Définition 1.2.1. [2]
Une fonction noyau $K\left(\frac{x_i - x_0}{h}\right) = K(u)$ vérifie les propriétés suivantes :

1. $K(u) \geq 0$.
2. $K(u)$ est normalisée de sorte que $\int K(u)du = 1$.
3. $K(u)$ atteint son maximum en 0 lorsque $x_i = x_0$ et décroît avec la distance $|x_0 - x_i|$.
4. $K(u)$ est symétrique : le noyau ne dépend que de la distance $|x_0 - x_i|$ et non du signe de $x_0 - x_i$.

Quelques fonctions noyaux usuelles

- Le noyau uniforme : $K(u) = \frac{1}{2}$, pour $u \in [-1, 1]$.
- Le noyau triangulaire : $K(u) = 1 - |u|$, pour $u \in [-1, 1]$.
- Le noyau quartic ou Bi-Weight : $K(u) = \frac{15}{16}(1 - u^2)^2$, pour $u \in [-1, 1]$.
- Le noyau Epanechnikov : $K(u) = \frac{3}{4}(1 - u^2)$, pour $u \in [-1, 1]$.
- Le noyau triweight : $K(u) = \frac{35}{32}(1 - u^2)^3$, pour $u \in [-1, 1]$.
- Le noyau normal : $K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right)$, pour $u \in]-\infty, +\infty[$.

Propriétés de l'estimateur

Proposition 2.[2]
Supposons que $m(.)$ et $f_X(.)$ sont de classe $\mathcal{C}^2(\mathbb{R})$ et que le noyau K est d'ordre 2, i.e. tel que :

$$\int_{\mathbb{R}} K(u)du = 1, \quad \int_{\mathbb{R}} uK(u)du = 0 \quad \text{et} \int_{\mathbb{R}} u^2 K(u)du < \infty.$$

Nous avons alors, lorsque $h \to 0$ et $nh \to \infty$

$$\begin{aligned} Biais(\hat{m}(x)) &= \mathbb{E}\left[\hat{m}(x) - m(x)\right] \\ &= \frac{h^2}{2}\left(m''(x) + 2m'(x)\frac{f_X'(x)}{f_X(x)}\right)\int u^2 K(u)du + o(h^2) \end{aligned}$$

Proposition 3. [2]

Supposons que $E[Y^2] < \infty$. À chaque point de continuité des fonctions $m(x)$, $f_X(x)$ et $\sigma^2(x)$, tel que $f_X(x) > 0$.

$$Var(\hat{m}(x)) = \mathbb{E}\left[(\hat{m}(x) - \mathbb{E}[\hat{m}(x)])^2\right] = \frac{1}{nh}\frac{\sigma^2(x)}{f_X(x)}\int K^2(u)du + o\left(\frac{1}{h}\right).$$

Remarque : Nous posons, par convenance

$$\sigma^2(x) = Var[Y|X = x],$$

lorsque cette expression est bien définie.

Le choix du paramètre de lissage

Le choix du paramètre de lissage h correspond à un arbitrage variance / biais :

- Plus h est élevé, plus la courbe $\hat{m}(x)$ sera lisse. La variance de l'estimation est limitée, mais l'estimateur $\hat{m}(x)$ peut être fortement biaisé.
- Plus h est faible, plus la courbe $\hat{m}(x)$ est irrégulière. Les biais d'estimation de $m(x)$ sont faibles, mais la variance de $\hat{m}(x)$ est très importante.

Le choix de h résulte donc d'un arbitrage biais versus variance, mais aussi d'un arbitrage lissage / non lissage de $m(x)$.

Chapitre 2

Régression non paramétrique par la méthode des fonctions splines

Le terme "fonction spline" a été introduit par Schoenberg en 1946, bien que leur origine puisse remonter aux travaux de Whittaker (1923) sur les méthodes de graduation de données. Cependant, dans le contexte mathématique, une spline est une courbe lisse définie par morceaux à l'aide de polynômes. Ces polynômes sont définis sur des intervalles spécifiques appelés "nœuds" qui sont reliés de manière continue (comme le montre la figure 2.1).

Dans les années 60 et 70, les splines se sont énormément développées et sont devenues importantes dans différents domaines tels que : les domaines mathématiques comme la théorie d'approximation, l'analyse numérique, la statistique et les domaines d'application comme l'animation, le graphisme et la topographie.

En 1990, les techniques spline sont intégrées dans des logiciels statistiques comme R pour faciliter leur utilisation pratique et aujourd'hui, les splines de lissage sont largement utilisées dans le Data Science et l'intelligence artificielle(IA).

Ce chapitre est consacré à la représentation de la notion des fonctions splines et l'estimation de la courbe de régression en utilisant la méthode des fonctions splines de lissage.

I.Généralités sur les fonctions splines

2.1 Définition

La fonction $f : \mathbb{R} \to \mathbb{R}$ est une spline de degré $(k-1)$(ou d'ordre k) avec des nœuds $x_1, ...x_n$ si :

(i). $f \in \mathcal{C}^{(k-2)}(\mathbb{R})$ ie : f est dérivable et ses dérivées sont continues jusqu'à l'ordre $k-2$.

(ii). $f(x) = P_i(x)$ pour tout $i = 1, ..., n-1$, où $P_i(x)$ est un polynôme de degré k sur le sous intervalle $[x_i, x_{i+1}]$ qui vérifie les conditions d'interpolations $P_i(x_i) = y_i$ et $P_i(x_{i+1}) = y_{i+1}$ pour tout $i = 1, ..., n-1$

Cet ensemble de fonctions est noté $S_k(x_1....x_n)$. Il contient l'ensemble des polynômes de degré inférieur ou égal à $k-1$.

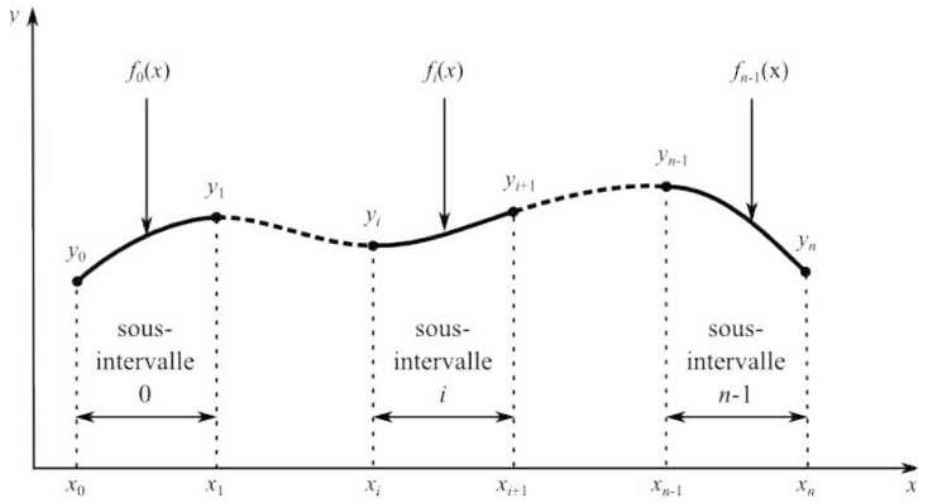

FIGURE 2.1 – Représentation de la spline

2.2 Différents types de splines

Il existe plusieurs catégories de splines, chacune ayant ses propres caractéristiques. Parmi elles :

2.2.1 Spline naturelle

Les splines peuvent avoir différentes conditions aux limites, et l'une de ces conditions est appelée "spline naturelle". Cette dernière impose que les dérivées secondes aux extrémités de la courbe soient fixées à zéro. Elle est souvent utilisée pour interpoler des données tout en minimisant la courbure.

Définition 2.1 : Soit$\mathbb{I} = [a; b]$ un intervalle de $\mathbb{R}$ tel que $a < x_1 < \ldots < x_n < b$. Une fonction spline f d'ordre $2k$ ou de degré impair $(2k-1)$ avec des points de nœuds $x_1, \ldots, x_n$ est dite naturelle, si elle est équivalente à un polynôme de degré inférieur ou égale à $(k-1)$ en dehors de l'intervalle $[x_1, x_n]$.

L'espace des fonctions splines naturelles d'ordre $2k$ est noté par $S_{2k}(x_1 \ldots x_n)$, cela signifie que toute fonction f appartenant à cet espace satisfait les conditions suivantes :

i. f est une spline de degré $2k-1$
ii. $f''(a) = f''(b) = 0$

2.2.2 Spline d'Hermite

Définition 2.2 : Une spline d'Hermite notée H définie sur un intervalle $[x_1, x_n]$ est une spline cubique de classe $\mathcal{C}^1$. Elle utilise des points de contrôle y_i qui influent sur la direction et la courbure locale d'une courbe ainsi que des vecteurs de tangente associés à ces points (voir la figure 2.2). Cela permet un meilleur contrôle de la forme de la spline. Sa construction est basée sur les fonctions de base locales qui s'appellent *"polynômes d'Hermite "*, ce qui permet d'assurer une interpolation lisse et continue entre ces points de contrôle et leurs tangentes.

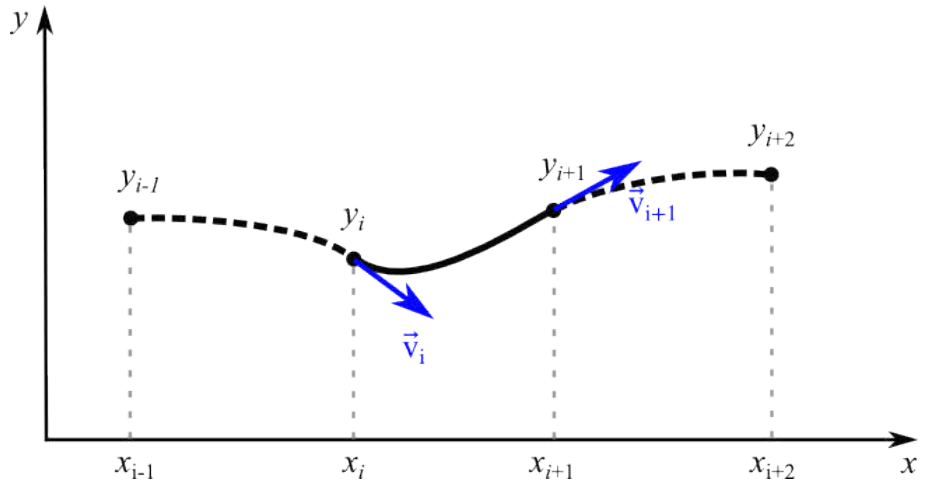

FIGURE 2.2 – Exemple d'une spline de Hermite

Mathématiquement, la spline d'Hermite est définie comme une combinaison linéaire de polynômes d'Hermite locaux telle que :

$$H(x) = \sum_{i=1}^{n} P_i(x) w_i$$

où :

- Les P_i sont des polynômes d'Hermite définis sur l'intervalle $[x_i, x_{i+1}]$ par :

  $$P_i(x) = (2t_i^3 - 3t_i^2 + 1)y_i + (-2t_i^3 + 3t_i^2)y_{i+1} + (x - x_i)(t_i^2 - t_i^2 + t_i)m_i + (x_{i+1} - x)(t_i^3 - t_i^2)m_{i+1}$$

  $$\forall i = 1, \ldots, n$$

  avec :

  - $t_i = \frac{x - x_i}{x_{i+1} - x_i}$ est le paramètre de normalisation qui appartient à l'intervalle $[0, 1]$.
  - m_i et m_{i+1} sont des dérivées premières aux points de contrôles.
- w_i sont les poids ou coefficients associés à chaque polynôme d'Hermite local $P_i(x)$.

Écriture matricielle

Il est fréquent d'adopter une représentation matricielle pour définir les polynômes d'une spline.

Cette matrice est appelée la matrice de base d'Hermite, où chaque ligne correspond au développement des coefficients w_i et chaque colonne permet d'extraire une fonction de base d'Hermite.

Supposons que pour i fixé nous ayons deux points de contrôle y_i et y_{i+1} avec des pentes de tangentes associées m_i et m_{i+1}.
La representation matricielle de polynôme d'Hermite peut être donnée sous la forme de produit matriciel :

$$P_i(x) = \begin{bmatrix} 1 & t_i & t_i^2 & t_i^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} y_i \\ m_i \\ y_{i+1} \\ m_{i+1} \end{bmatrix}$$

$$\Rightarrow P_i(x) = h_0(t_i).y_i + h_1(t_i).y_{i+1} + h_2(t_i).m_i + h_3(t_i).m_{i+1}$$

avec

$$\begin{cases} h_0(t_i) = 2t_i^3 - 3t_i^2 + 1 \\ h_1(t) = -2t_i^3 + 3t_i^2 \\ h_2(t_i) = t_i^3 - 2t_i^2 + t_i \\ h_3(t_i) = t_i^3 - t_i^2 \end{cases}$$

Remarque : La représentation matricielle offre une manière efficace de calculer les polynômes d'Hermite dans le contexte des splines, en minimisant le nombre d'opérations nécessaires lors du tracé.

2.2.3 Spline de Catmull-Rom

Définition 2.3 : Une spline de Catmull-Rom est une courbe paramétrique cubique définie par quatre points de contrôle y_{i-1}, y_i, y_{i+1} et y_{i+2} et elle est calculée pour les segments de courbe allant de y_i à y_{i+1}(voir la figure 2.3). La spline de Catmull-Rom est utilisée dans l'animation et le graphisme par ordinateur pour créer des courbes douces qui passe exactement par ces points de contrôle.

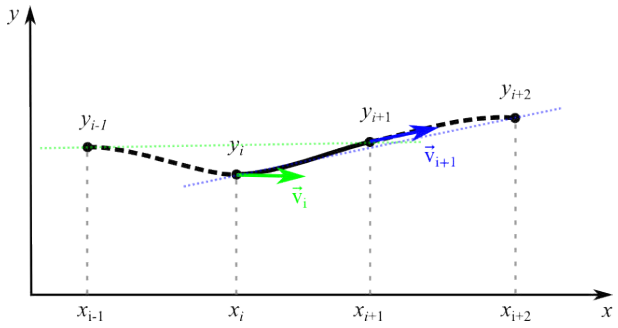

FIGURE 2.3 – Courbe représentative d'une spline de Catmull-Rom

La formule générale d'un polynôme cubique d'une spline de Catmull-Rom avec le paramètre t dans l'intervalle$[0, 1]$ est la suivante :

$$f_i(t) = \frac{1}{2}[(-y_{i-1}+3y_i-3y_{i+1}+y_{i+2})t^3+(2y_{i-1}+5y_i+4y_{i+1}-y_{i+2})t^2+(y_{i-1}+y_{i+1})t+2y_i]$$

$$\forall i = 1, ..., n-2$$

Ces polynômes peuvent s'écrire sous la formule matricielle :

$$f_i(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \cdot \frac{1}{2} \begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} y_{i-1} \\ y_i \\ y_{i+1} \\ y_{i+2} \end{bmatrix}$$

2.2.4 Spline de Bézier

Définition 2.4 : Une spline de Bézier de degré n est une courbe paramétrique, caractérisée par $n+1$ points de contrôle. Elle est largement utilisée dans le design graphique et elle permet une manipulation intuitive de la forme de la courbe. Mathématiquement parlant,une spline de Bézier est définie par :

$$B(t) = \sum_{i=0}^{n} B_i^n(t) y_i \quad \forall t \in [0,1] \quad \text{avec} \quad y_i = f(x_i)$$

Où :

- y_i sont des points de contrôle
- $B_i^n(t)$ sont des polynômes de Bernstein définis par :

$$B_i^n(t) = \binom{n}{i}.(1-t)^{n-i}.t^i = \frac{n!}{i!(n-i)!}.(1-t)^{n-i}.t^i \quad \forall t \in [0,1]$$

L'image ci-dessous illustre des exemples de construction de la courbe de Bézier pour différentes valeurs de t :

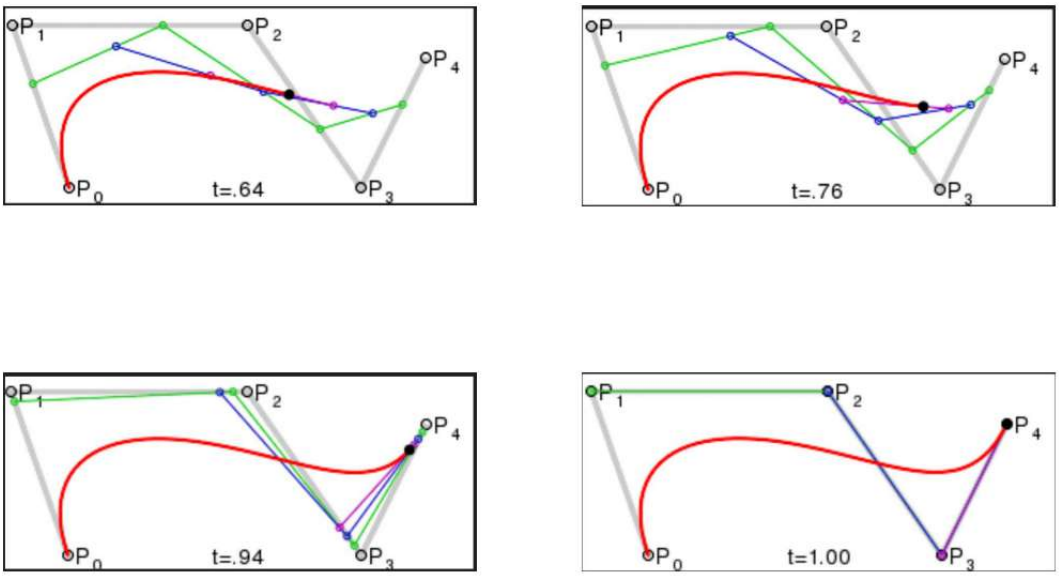

FIGURE 2.4 – Exemples de construction de courbe de Bézier

Pour $n = 3$, on parle de la spline de Bézier cubique(voir la figure 2.5) telle que :

$$B(t) = \sum_{i=0}^{3} B_i^3(t) y_i = (1-t)^3.y_0 + 3(1-t)^2.t.y_1 + 3(1-t).t^2.y_2 + t^3.y_3 \quad \forall t \in [0,1]$$

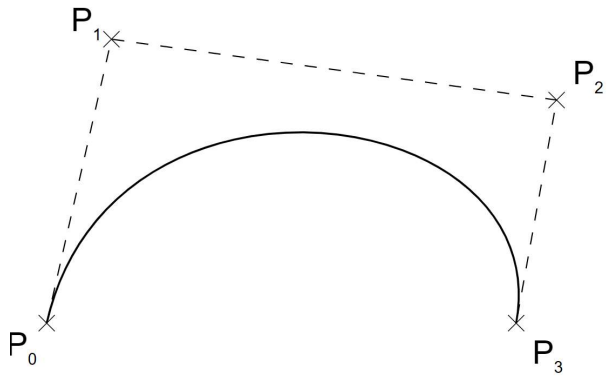

FIGURE 2.5 – Courbe représentative d'une spline de Bézier cubique

2.2.5 Fonction B-Spline

Le terme «B-spline» provient de l'expression anglaise «Basis spline», introduite pour la première fois par le mathématicien Isaac Jacob Schoenberg en 1946 dans le cadre de ses travaux de recherche sur les approximations de fonctions. Dans les années 1960 et 1970, les B-splines ont été développées et popularisées pour leur utilisation en conception assistée par ordinateur(CAO).

En 1966, Curry et Schoenberg ont démontré que toute fonction spline $f(x)$ d'ordre k peut être représentée par une combinaison linéaire des fonctions B-spline

$$f(x) = \sum_{i=1}^{z} a_i B_i^k(x), \forall x \in [a, b]$$

Définition 2.5 : La fonction de base B-spline de degré k est définie de manière récursive de base B-spline de degré $k-1$.

Soit $x_1 = ... = x_k < x_{k+1} .... < x_z ... < x_{z+1} = ... = x_{z+k}$ une suite de nœuds sur la droite réelle. La fonction de base B-spline de degré k est définie par :

- Si $k = 0$

$$B_i^0(x) = \begin{cases} 1 & \text{si } x_i \leq x < x_{i+1} \\ 0 & \text{sinon} \end{cases}$$

- Si $k \geq 1$

$$B_i^k(x) = w_{i,k}(x) B_i^{k-1}(x) + (1 - w_{i+1,k}(x)) B_{i+1}^{k-1}(x)$$

  Où :

$$w_{i,k}(x) = \begin{cases} \frac{x - x_i}{x_{i+k} - x_i} & \text{si } x_i < x_{i+1} \\ 0 & \text{si } x_i = x_{i+1} \end{cases}$$

Remarque : Par convention, lorsque les nœuds sont confondus, nous supposons qu'une fraction dont le dénominateur est nul est égale à zéro.

La figure ci-dessous montre un exemple de B-splines de degré 3 relatives au vecteur de noeuds $(-1, 0, 0, 0, 0, 1, 2, 3..)$:

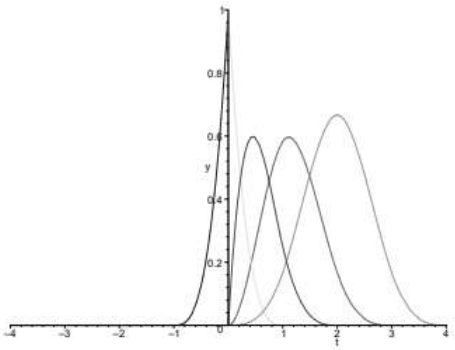

FIGURE 2.6 – Exemple des B-splines de degré 3

Quelques propriétés

- **Caractéristiques fondamentales**

Proposition 1 :

1. B_j^k est un polynôme de degré inférieur ou égale à k sur chaque intervalle $[x_i, x_{i+k}[$.
2. $B_i^k(x) = 0 \quad x < x_i \quad ou \quad x \geq x_{i+k}$.
3. $$B_{i,k}(x) = \begin{cases} 0 & \text{si } x \notin \{x_{i+1}, \ldots, x_{i+k}\} \\ 1 & \text{si } x = x_{i+1} = \ldots = x_{i+k} < x_{i+k+1} \end{cases}$$
4. $0 < B_i^k(x) \leq 1 \quad x \in]x_i, x_{i+k}[$.

- **Symétries**

Proposition 2 : Supposons que le vecteur de nœuds x est périodique,i.e, il existe un entier I et un réel T tels que : pour tout $i \in \mathbb{Z}$

$$x_{i+\mathrm{I}} = x_i + \mathrm{T}$$

Dans ce cas, les fonctions B-splines correspondantes satisfont l'égalité suivante :

$$B_i^k(x - \mathrm{T}) = B_{i+\mathrm{I}}^k(x) \quad \forall i \in \mathbb{Z}$$

Preuve : (Par récurrence sur k)
Pour $k = 0$, soit $B_i^0(x - \mathrm{T})$ la fonction caractéristique de l'intervalle $[x_i - \mathrm{T}, x_{i+1} + \mathrm{T}[= [x_{i+\mathrm{I}}, x_{i+\mathrm{I}+1}[$ telle que :

$$B_i^0(x - \mathrm{T}) = \begin{cases} 1 & \text{si } x \in [x_{i+\mathrm{I}}, x_{i+\mathrm{I}+1}[\\ 0 & \text{sinon} \end{cases}$$

$$\text{alors,} \quad B_i^0(x - \mathrm{T}) \quad \text{coïncide avec} \quad B_{i+\mathrm{I}}^0(x).$$

D'autre part,

$$w_{i,0}(x - \mathrm{T}) = w_{i+\mathrm{I},0}(x) \quad \forall i \in \mathbb{Z}.$$

Supposons que, $B_i^{k-1}(x - \mathrm{T}) = B_{i+\mathrm{I}}^{k-1}(x)$ est vraie pour $k-1$ et montrons qu'elle est vraie pour k,

$$\begin{aligned} B_i^k(x - \mathrm{T}) &= w_{i,k}(x - \mathrm{T}) B_i^{k-1}(x - \mathrm{T}) + (1 - w_{i+1,k}(x - \mathrm{T})) B_{i+1}^{k-1}(x - \mathrm{T}) \\ &= w_{i+\mathrm{I},k}(x) B_{i+\mathrm{I}}^{k-1}(x) + (1 - w_{i+\mathrm{I}+1,k}(x)) B_{i+\mathrm{I}+1}^{k-1}(x) \\ &= B_{i+\mathrm{I}}^k(x). \end{aligned}$$

$$\text{Donc,} \quad B_i^k(x - \mathrm{T}) = B_{i+\mathrm{I}}^k(x) \quad \forall i \in \mathbb{Z}.$$

Proposition 3 : Supposons que le vecteur de nœuds x est symétrique à un réel a,i.e, il existe des entiers I et ϵ(qui détermine la largeur de la symétrie) tels que : pour tout $i = \frac{1}{2}(I - \epsilon), ..., \frac{1}{2}(I + \epsilon)$,

$$x_{\mathrm{I}-i} = 2a - x_i.$$

Alors, les fonctions B-splines correspondantes satisfont :

$$B_i^k(2a-x) = B_{\mathrm{I}-i-k-1}^k(x) \quad \forall i = \frac{1}{2}(I-\epsilon), ..., \frac{1}{2}(I+\epsilon).$$

Preuve :(Par récurrence sur k)
Pour $k=0, \quad i = \frac{1}{2}(I-\epsilon), ..., \frac{1}{2}(I+\epsilon)-1$ la fonction $x \to B_i^0(2a-x)$ est la fonction caractéristique de l'intervalle $]2a-x_{i+1}, 2a-x_i] =]x_{\mathrm{I}-i-1}, x_{\mathrm{I}-i}]$, qu'elle coïncide avec $B_{\mathrm{I}-i-1}^0(x)$.

D'autre part,

$$w_{i,0}(2a-x) = 1 - w_{\mathrm{I}-i,0}(x) \quad \forall i = \frac{1}{2}(I-\epsilon), ..., \frac{1}{2}(I+\epsilon)-1$$

Supposons que pour tout $i = \frac{1}{2}(I-\epsilon), ..., \frac{1}{2}(I+\epsilon)-k-2$,
$B_i^{k-1}(2a-x) = B_{\mathrm{I}-i-k}^{k-1}(x)$ est vraie pour $k-1$ et montrons qu'elle est vraie pour k.

On a :

$$\begin{aligned} B_i^k(2a-x) &= w_{i,k}(2a-x)B_i^{k-1}(2a-x) + (1-w_{i+1,k}(2a-x))B_{i+1}^{k-1}(2a-x) \\ &= w_{\mathrm{I}-i-k,k}(x)B_{\mathrm{I}-i-k}^{k-1}(x) + (w_{\mathrm{I}-i-k-1,k}(x))B_{\mathrm{I}-i-k-1}^{k-1}(x) \\ &= B_{\mathrm{I}-i-k-1}^k(x), \end{aligned}$$

pour tout $i = \frac{1}{2}(I-\epsilon), \ldots, \frac{1}{2}(I+\epsilon)-k-1$

Remarque : Nous observons la symétrie des fonctions B-splines dans deux cas particuliers : Les B-splines uniformes et les polynômes de Bernstein.

- **B-splines uniformes :** Supposons que, pour $i \in \mathbb{Z}$, $x_i = i$. Alors, pour tout $i \in \mathbb{Z}$,$k \geq 1$ et $x \in \mathbb{R}$

$$B_i^k(x+1) = B_{i-1}^k(x) \quad et \quad B_0^k(k+1-x) = B_0^k(x).$$

- **Polynôme de Bernstein :** On pose $x_0 = \ldots = x_k = 0$, $x_{k+1} = \ldots = x_{2k+1} = 1$. Alors pour tout $i = 0, \ldots, k$,

$$B_i^k(x) = \begin{cases} \binom{k}{i} x^i (1-x)^{k-i} & \text{si } x \in [0,1[\\ 0 & \text{sinon} \end{cases}$$

- **Partition de l'unité**

Proposition 4 : Soit $B_i^k(x)$ une B-spline d'ordre k définie sur l'intervalle $[x_0, x_{n+k}]$ avec des nœuds $x_0 \leq \ldots \leq x_{n+k}$. Pour $m \geq 2k$, sur l'intervalle $[x_k, x_{m-k}[$:

$$\sum_{i=0}^{m-k-1} B_i^k(x) = 1.$$

Preuve : Adoptons le raisonnement par récurrence sur k.
Pour $k = 0$, elle est clairement vraie.
Supposons que la propriété est vraie pour $k-1$, puis on montre qu'elle est également vraie pour k.
on a :

$$\begin{aligned}\sum_{i=0}^{m-k-1} B_i^k(x) &= [w_{0,k}(x)B_0^{k-1}(x) + (1 - w_{1,k}(x))B_1^{k-1}(x)] \\ &+ [w_{1,k}(x)B_1^{k-1}(x) + (1 - w_{2,k}(x))B_2^{k-1}(x)] \\ &+ \ldots + [w_{m-k-2,k}(x)B_{m-k-2}^{k-1}(x) + (1 - w_{m-k-1,k}(x))B_{m-k-1}^{k-1}(x)] \\ &+ [w_{m-k-1,k}(x)B_{m-k-1}^{k-1}(x) + (1 - w_{m-k,k}(x))B_{m-k}^{k-1}(x)] \\ &= (w_{0,k}(x) - 1)B_0^{k-1}(x) + \sum_{i=0}^{m-k} B_i^{k-1}(x) - w_{m-k,k}(x)B_{m-k}^{k-1}(x).\end{aligned}$$

D'après l'hypothèse de récurrence, $\sum_{i=0}^{m-k} B_i^{k-1}(x) = 1$ sur l'intervalle $[x_{k-1}, x_{m-k-1}[$ par conséquent sur l'intervalle $[x_{k-1}, x_{m-k-1}[$.

Comme on a d'après la proposition 1, les fonctions B_0^{k-1} et B_{m-k}^{k-1} sont respectivement nulles en dehors des l'intervalles $[x_0, x_k[$ et $[x_{m-k}, x_m[$.

Donc,

$$\sum_{i=0}^{m-k-1} B_i^k(x) = 1 \quad sur \quad [x_k, x_{m-k}[$$

- **Continuité**

Rappel : En rappelant que si la fonction f définie sur l'ensemble des nombres réels $\mathbb{R}$ admet des limites à droite notées $f(x^+)$ et à gauche notées $f(x^-)$.

Lemme 1 : Si $x_{i-1} < x_i = x_{i+1} = \ldots x_{i+r-1} < x_{i+r}$, alors $B_{i-1}^r(x_i^{\pm}) = 1$. Par consequent, toutes les fonctions B_j^k pour $k \geq r$ sont continues en x_i, pour tout $j = i-1, \ldots, i-1+k$.

Preuve : Montrons par récurrence sur $k \leq r$ que :

$$B_{i-1}^r(x_i^{\pm}) = B_{i-1}^{r-k}(x_i^{\pm}) + B_i^{r-k}(x_i^{\pm}) + \ldots + B_{i-1+k}^{r-k}(x_i^{\pm}).$$

Pour $k = 0$,l'énonce est clairement vrai.
Supposons qu'il est prouvé pour un certain k. Nous avons pour tout $j = i-1, \ldots, i-1+k$,

$$B_j^{r-k}(x_i^{\pm}) = w_{j,r-k}(x_i)B_j^{r-k-1}(x_i^{\pm}) + (1 - w_{j+1,r-k}(x_i))B_{j+1}^{r-k-1}(x_i^{\pm}).$$

Or

$$w_{j,r-k}(x_i) = \begin{cases} 1 & \text{si } j = i-1 \\ 0 & \text{sinon} \end{cases}$$

Alors,

$$B_{i-1}^{r-k}(x_i^{\pm}) = B_{i-1}^{r-k-1}(x_i^{\pm}) + B_i^{r-k-1}(x_i^{\pm})$$

et pour $j = i, i+1, \ldots, i-1+k$,

$$B_j^{r-k}(x_i^{\pm}) = B_{j+1}^{r-k-1}(x_i^{\pm}).$$

En effectuant cette sommation, on obtient :

$$B_{i-1}^{r}(x_i^{\pm}) = B_{i-1}^{r-k}(x_i^{\pm}) + B_i^{r-k}(x_i^{\pm}) + \ldots + B_{i-1+k}^{r-k}(x_i^{\pm}).$$

Pour $k = 0$, nous avons :

$$B_j^0(x_i^+) = \begin{cases} 0 & \text{si } j = i-1, \ldots, i+r-2 \\ 1 & \text{si} j = i+r-1 \end{cases}$$

$\Rightarrow$ la somme $B_{i-1}^r(x_i^+) = 1$, alors B_{i-1}^r est continue en x_i^+.

$$B_j^0(x_i^-) = \begin{cases} 0 & \text{si } j = i, \ldots, i+r-1 \\ 1 & \text{si } j = i-1 \end{cases}$$

$\Rightarrow$ la somme $B_{i-1}^r(x_i^-) = 1$, alors B_{i-1}^r est continue en x_i^-.

Ainsi, B_{i-1}^r est continue en x_i.

Puisque les B-splines de degré r sont bornées dans l'intervalle $[0, 1]$ et leur somme vaut 1, cela implique que toutes ces fonctions sont continues en x_i. L'hypothèse de la récurrence définissant les B-splines prouve que les B-splines de degré supérieur ou égale à r sont également continues en x_i.

- **Différentiabilité**

Lemme 2 : Pour tout $k \geq 0$, la fonction B_i^k est dérivable à droite, de dérivée :

$$\begin{aligned} B_i^{k\prime}(x) &= \frac{k}{x_{i+k} - x_i} B_i^{k-1}(x) - \frac{k}{x_{i+1+k} - x_{i+1}} B_{i+1}^{k-1}(x) \\ &= k(w_{i,k}^{'}(x) B_i^{k-1}(x) - w_{i+1,k}^{'}(x) B_{i+1}^{k-1}(x)) \end{aligned}$$

Preuve : (raisonnement par récurrence sur k)
L'énoncé est vrai pour $k = 0$.
Supposons qu'il est démontré pour un certain degré inférieur ou égale à $k-1$ et on montre est aussi vrai pour k.
On sait que :

$$\begin{aligned} B_i^{k\prime}(x) &= w_{i,k}^{'}(x) B_i^{k-1}(x) + w_{i,k}(x) B_i^{k-1\prime}(x) - w_{i+1,k}^{'}(x) B_{i+1}^{k-1}(x) + (1 - w_{i+1,k}(x)) B_{i+1}^{k-1\prime}(x) \\ &= w_{i,k}^{'}(x) B_i^{k-1}(x) - w_{i+1,k}^{'}(x) B_{i+1}^{k-1}(x) + w_{i,k}(x) B_i^{k-1\prime}(x) + (1 - w_{i+1,k}(x)) B_{i+1}^{k-1\prime}(x). \end{aligned}$$

D'après l'hypothèse de récurrence on a ,

$$\begin{aligned}\frac{1}{k-1}(w_{i,k}(x)B_i^{k-1\prime}(x)+(1-w_{i+1,k}(x))B_{i+1}^{k-1\prime}(x)) &= w_{i,k}(x)w_{i,k-1}^{'}(x)B_i^{k-2}(x)\\ &- w_{i,k}(x)w_{i+1,k-1}^{'}(x)B_{i+1}^{k-2}(x)\\ &+ (1-w_{i+1,k}(x))w_{i+1,k-1}^{'}(x)B_{i+1}^{k-2}(x)\\ &- (1-w_{i+1,k}(x))w_{i+2,k-1}^{'}(x)B_{i+2}^{k-2}(x).\end{aligned}$$

Or :

$$w_{i,k}(x)w_{i,k-1}^{'}(x)=\frac{x-x_i}{x_{i+k}-x_i}\frac{1}{x_{i+k-1}-x_i}=\frac{x-x_i}{x_{i+k-1}-x_i}\frac{1}{x_{i+k}-x_i}=w_{i,k-1}(x)w_{i,k}^{'}(x).$$

$$(1-w_{i+1,k}(x))w_{i+2,k-1}^{'}(x)=\frac{x_{i+1+k}-x}{x_{i+1+k}-x_{i+1}}\frac{1}{x_{i+k+1}-x_{i+2}}=(1-w_{i+2,k-1}(x))w_{i+1,k}^{'}(x).$$

Et

$$\begin{aligned}&-w_{i,k}(x)w_{i+1,k-1}^{'}(x)+(1-w_{i+1,k}(x))w_{i+1,k-1}^{'}(x)\\ &\quad=\frac{x_i-x}{(x_{i+k}-x_i)(x_{i+k}-x_{i+1})}+\frac{x_{i+k+1}-x}{(x_{i+k+1}-x_{i+1})(x_{i+k}-x_{i+1})}\\ &\quad=\frac{x_{i+k}-x-(x_{i+k}-x_i)}{(x_{i+k}-x_i)(x_{i+k}-x_{i+1})}+\frac{x_{i+1}-x+(x_{i+k+1}-x_{i+1})}{(x_{i+k+1}-x_{i+1})(x_{i+k}-x_{i+1})}\\ &\quad=\frac{1}{(x_{i+k}-x_{i+1})}\left(\frac{x_{i+k}-x}{x_{i+k}-x_i}-1+\frac{x_{i+1}-x}{x_{i+k+1}-x_{i+1}}+1\right)\\ &\quad=\frac{x_{i+k}-x}{(x_{i+k}-x_i)(x_{i+k}-x_{i+1})}+\frac{x_{i+1}-x}{(x_{i+k+1}-x_{i+1})(x_{i+k}-x_{i+1})}\\ &\quad=-w_{i+1,k-1}(x)w_{i+1,k}^{'}(x)+(1-w_{i+1,k-1}(x))w_{i,k}^{'}(x).\end{aligned}$$

Alors,

$$\begin{aligned}\frac{1}{k-1}(w_{i,k}(x)B_i^{k-1\prime}(x)+(1-w_{i+1,k}(x))B_{i+1}^{k-1\prime}(x)) &= w_{i,k-1}(x)w_{i,k}^{'}(x)B_i^{k-2}(x)\\ &- w_{i+1,k-1}(x)w_{i+1,k}^{'}(x)B_{i+1}^{k-2}(x)\\ &+ (1-w_{i+1,k-1}(x))w_{i,k}^{'}(x)B_{i+1}^{k-2}(x)\\ &- (1-w_{i+2,k-1}(x))w_{i+1,k}^{'}(x)B_{i+2}^{k-2}(x)\\ &= w_{i,k}^{'}(x)(w_{i,k-1}(x)B_i^{k-2}(x)\\ &+ (1-w_{i+1,k-1}(x))B_{i+1}^{k-2}(x))\\ &- w_{i+1,k}^{'}(x)(w_{i+1,k-1}(x)B_{i+1}^{k-2}(x)\\ &+ (1-w_{i+2,k-1}(x))B_{i+1}^{k-2}(x))\\ &= w_{i,k}^{'}(x)B_i^{k-1}(x)-w_{i+1,k}^{'}(x)B_{i+1}^{k-1}(x).\end{aligned}$$

Ainsi,

$$B_i^{k\prime}(x)=k(w_{i,k}^{'}(x)B_i^{k-1}(x)-w_{i+1,k}^{'}(x)B_{i+1}^{k-1}(x))$$

Proposition 5 :

1. La fonction B_i^k est de classe $\mathcal{C}^\infty$ à droite de chaque point.
2. Au voisinage d'un nœud de multiplicité r(i.e il apparaît r fois dans la séquence des nœuds), la fonction B_i^k est seulement de classe $\mathcal{C}^{k-r}$

Preuve :

1. Le lemme ci-dessus établit l'existence de dérivées à droite de tous ordres ce qui implique que la B-spline de degré k est de classe $\mathcal{C}^\infty$ à droite de chaque point.
2. Si x est un nœud de multiplicité r, alors les B-splines de degré r sont continues en x et d'après le lemme 2 les B-splines d'ordre $r+1$ sont de classe $\mathcal{C}^1$ au voisinage de x. En appliquant le raisonnement par récurrence, on démontre que B_i^k avec $k \geq r$ sont de classe $\mathcal{C}^{k-r}$ au voisinage de x.

2.2.6 Fonction spline cubique

Soient $\mathbb{I} = [a, b]$ un intervalle de $\mathbb{R}$ et $a < x_1 < ..... < x_n < b$ une partition de l'intervalle $\mathbb{I}$.
Une spline cubique f définie sur$[x_i, x_{i+1}]$ par :

$$f(x) = \sum_{i=1}^{n} P_i(x) \mathbb{1}_{[x_i, x_{i+1}[}(x)$$

avec les propriétés suivantes :

1. Sur chaque intervalle $[x_i, x_{i+1}[$, P_i est un polynôme de degré 3 (polynôme cubique) :

$$P_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i, \forall i = 1, ..., n$$

2. La fonction f est différentiable deux fois et ses dérivées (la première et la seconde dérivée) sont continues sur l'intervalle $[a, b]$.

Principales propriétés

- $(C_1) : P_i(x_i) = y_i$, $\forall i = 1, ..., n$.
  Cette condition assure que la spline f passe exactement par les points donnés.

- $(C_2) : P'_{i-1}(x_i) = P'_i(x_i)$.
  Cette propriété garantit que les pentes de la spline sont continues aux points de jonction.

- $(C_3) : P''_{i-1}(x_i) = P''_i(x_i)$.
  Cette propriété impose que la courbure de la spline est continue aux points de jonction.

- $(C_4) : P''_1(a) = P''_n(b) = 0$.
  Les dérivées secondes aux extrémités sont nulles, ce qui donne une courbe lisse.

Construction de spline cubique

On sait que les coefficients de la spline cubique a_i,b_i,c_i et d_i sont déterminés de manière à satisfaire les conditions d'interpolation et de continuité. Alors, comment peut-on les choisir ?

Dans toute cette construction, on pose $z_{i-1} = x_i - x_{i-1}$ et $y_i = f(x_i)$.
Supposons que f une spline cubique satisfaisant toutes les conditions définies précédemment, on a :

$$P_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \tag{2.1}$$

$$\forall x \in [x_i, x_{i+1}[,\ \forall i = 1, ..., n$$

La première et la seconde dérivées de P_i :
Pour tout $i = 1, ..., n$

$$P_i^{'}(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i, \tag{2.2}$$

$$P_i^{''}(x) = 6a_i(x - x_i) + 2b_i. \tag{2.3}$$

On sait que :

$$P_i(x_i) = a_i(x_i - x_i)^3 + b_i(x_i - x_i)^2 + c_i(x\_x_i) + d_i.$$
$$\Rightarrow P_i(x_i) = d_i.$$

Et d'après la propriété (C_1), on a :

$$P_i(x_i) = y_i.$$

Comme f doit être continue sur tout son l'intervalle, on peut déduire que chaque sous fonction doit s'aligner avec les points de données, alors :

$$P_{i-1}(x_i) = P_i(x_i) \Leftrightarrow a_{i-1}(x_i - x_{i-1})^3 + b_{i-1}(x_i - x_{i-1})^2 + ci - 1(x\_x_{i-1}) + d_{i-1} = y_i. \tag{2.4}$$

On remplace par z_{i-1} dans (2.4), on obtient :

$$y_i = a_{i-1}z_{i-1}^3 + b_{i-1}z_{i-1}^2 + c_{i-1}z_{i-1} + d_{i-1} \tag{2.5}$$

donc,

$$y_i = d_i$$

À partir de (C_2), on a :

$$P_{i-1}^{'}(x_i) = P_i^{'}(x_i).$$

D'après l'équation (2.2) on peut écrire :

$$P_i^{'}(x_i) = c_i$$

$$P_{i-1}^{'}(x_i) = P_i^{'}(x_i) \Leftrightarrow 3a_{i-1}(x_i - x_{i-1})^2 + 2b_{i-1}(x_i - x_{i-1}) + c_{i-1} = c_i. \tag{2.6}$$

Remplaçant par z_{i-1} dans (2.6), on aura :

$$3a_{i-1}z_{i-1}^2 + 2b_{i-1}z_{i-1} + c_{i-1} = c_i. \tag{2.7}$$

Par l'équation (2.3) on obtient :

$$P_i^{''}(x_i) = 6a_i(x_i - x_i) + 2b_i = 2b_i$$

Et par la condition (C_3) nous avons :

$$P_{i-1}^{''}(x_i) = P_i^{''}(x_i) \Leftrightarrow 6a_{i-1}z_{i-1} + 2b_{i-1} = 2b_i. \tag{2.8}$$

La condition que les dérivées secondes aux extrémités doivent être nulles est exprimée par :

$$P_1^{''}(a) = P_n^{''}(b) = 0 \Leftrightarrow d_1 = d_n = 0.$$

L'équation (2.5) donne :

$$c_{i-1} = \frac{y_i - y_{i-1}}{z_{i-1}} - a_{i-1}z_{i-1}^2 - b_{i-1}z_{i-1} \tag{2.9}$$

avec

$$y_{i-1} = d_{i-1}.$$

L'équation(2.8) donne :

$$a_{i-1} = \frac{b_i - b_{i-1}}{3z_{i-1}}. \tag{2.10}$$

Suppléant (2.10) dans (2.9) on aura :

$$c_{i-1} = \frac{y_i - y_{i-1}}{z_{i-1}} - \frac{b_i z_{i-1} + b_{i-1}z_{i-1}}{3} - b_{i-1}z_{i-1} = \frac{y_i - y_{i-1}}{z_{i-1}} - \frac{1}{3}(b_i z_{i-1} + 2b_{i-1}z_{i-1}). \tag{2.11}$$

Remplaçons les équations (2.10) et (2.11) dans (2.7) :

$$\frac{1}{3}(b_{i-1}z_{i-1} + 2b_i(z_{i-1} + z_i) + b_{i+1}z_i) = y_{i-1}\frac{1}{z_{i-1}} - y_i(\frac{1}{z_i} + \frac{1}{z_{i-1}}) + y_{i+1}\frac{1}{z_i} \tag{2.12}$$

Alors :

$$\begin{cases} b_i = \frac{1}{z_{i-1}^2}\left(\frac{1}{6}c_i + \frac{1}{3}c_{i+1}\right) - \frac{z_{i-1}}{6}(y_i - y_{i+1}), \\ a_i = \frac{1}{6z_{i-1}}(c_{i+1} - c_i), \\ d_i = \frac{1}{2z_{i-1}}\left(\frac{y_{i+1} - y_i}{z_{i-1}} - \frac{z_{i-1}}{6}(c_i + 2c_{i+1})\right). \end{cases}$$

Nous avons $P_i(x_i) = y_i$ et $P_i^{''}(x_i) = 2b_i$.

Notons par$F = (F_1 \dots F_n)^t$ et $\Gamma = (\gamma_2 \dots \gamma_{n-1})$ tels que : $F_i = f(x_i)$ et $\gamma_i = f^{''}(x_i)$ avec $\gamma_1 = \gamma_n = 0$.

La construction de la spline cubique implique par la résolution du système linéaire $A.C = \Gamma$, où :

- C est un vecteur de coefficients que nous cherchons à déterminer.
- A est une matrice tridiagonale de taille $n \times n$ définie comme suit :

$$A = \begin{bmatrix} R & Q^t \\ Q & 0 \end{bmatrix} = QR^{-1}Q^t$$

où,

- Q est une matrice de taille $n \times (n-2)$ de composantes q_{ij} :

$$q_{ij} = \begin{cases} \frac{1}{h_{j-1}} & \text{si } i = j-1 \\ -\left(\frac{1}{h_{j-1}} + \frac{1}{h_j}\right) & \text{si } i = j \\ \frac{1}{h_j} & \text{si } i = j+1 \\ 0 & \text{si } |i-j| \geq 2 \end{cases}$$

- R est une matrice symétrique de taille $(n-2)\times(n-2)$ de composantes r_{ij} :

$$r_{ij} = \begin{cases} \frac{1}{3}(h_{i-1} + h_i) & \text{si } j = i, i = 2, \ldots, n-1 \\ \frac{1}{6}h_i & \text{si } j = i+1, i = 2, \ldots, n-2 \\ 0 & \text{si } |i-j| \geq 2 \end{cases}$$

alors,

$$A.C = \Gamma \Leftrightarrow \begin{bmatrix} R & Q^t \\ Q & 0 \end{bmatrix} \cdot \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} 0 \\ \gamma_2 \\ \vdots \\ \gamma_{n-1} \\ 0 \end{bmatrix} \Leftrightarrow \begin{cases} r_{11}c_1 + r_{12}c_2 = 0 \\ q_{21}c_1 + r_{22}c_2 + q_{23}c_3 = \gamma_2 \\ q_{32}c_2 + r_{33}c_3 + q_{34}c_4 = \gamma_3 \\ q_{43}c_3 + r_{44}c_4 + q_{45}c_5 = \gamma_4 \\ \vdots \\ q_{n-2,n-3}c_{n-3} + r_{n-2,n-2}c_{n-2} + q_{n-1,n-1}c_{n-1} \\ = \gamma_{n-1} \\ q_{n-1,n-2}c_{n-2} + r_{n-1,n-1}c_{n-1} = 0 \end{cases}$$

Théorème 2. ***[2]***
On dit que les vecteurs F et Γ définissent une spline cubique si et seulement si ils satisfont :

$$Q^t F = R\Gamma.$$

Si cette relation est vérifiée, alors :

$$\int_a^b f''(x)^2 dx = \Gamma R \Gamma^t = F^t A F.$$

II.Interpolation

Le problème réside dans la détermination du comportement global de la fonction à partir d'un échantillon limité de ses valeurs. L'interpolation consiste à trouver une fonction qui passe exactement par ces points, permettant ainsi d'interpoler des valeurs inconnues à partir de données discrètes.

Mathématiquement parlant, soit $(x_1, y_1), ..., (x_n, y_n)$ un ensemble de points distincts où $x_1 < ... < x_n$. . Cela nous garantit l'unicité du polynôme d'interpolation.

L'interpolation consiste à reconstruire une fonction $f(x)$ à partir de ces points, i.e : $f(x_i) = y_i \quad \forall i = 1, ..., n$.

Abordons d'abord le cas le plus élémentaire. Comment procéder à l'interpolation entre deux points ayant pour coordonnées (x_1, y_1) et (x_2, y_2) ?

Pour cela nous pouvons utiliser la formule suivante :

$$f(x) = ax + b.$$

L'objectif est de déterminer les coefficients a et b pour ces deux points données. En résolvant le système suivant :

$$\begin{cases} y_1 = ax_1 + b \\ y_2 = ax_2 + b \end{cases}$$

alors,

$$\begin{cases} a = \frac{y_2 - y_1}{x_2 - x_1} \\ b = y_1 - ax_1 \end{cases}$$

Supposons que nous voulions interpoler trois points $(x_1, y_1), (x_2, y_2)$ et (x_3, y_3). Dans le cas où ils ne sont pas alignés, une droite simple ne suffit pas.Alors, nous devons utiliser une fonction polynomiale de degré supérieur.

$$f(x) = ax^2 + bx + c.$$

Il faut résoudre le système à trois équations à trois inconnues (a, b, c) suivant :

$$\begin{cases} y_1 = ax_1^2 + bx_1 + c \\ y_2 = ax_2^2 + bx_2 + c \\ y_3 = ax_3^2 + bx_3 + c \end{cases}$$

Comment peut-on réaliser l'interpolation pour un ensemble de n points ?

2.1 Interpolation polynomiale classique

2.1.1 Polynôme d'interpolation de Lagrange

Définition 2.1.1. On appelle polynôme de Lagrange que l'on note $l_i(x)$, les polynômes de degré inférieur ou égal à $(n-1)$ définis comme suit :

$$l_i(x) = \prod_{j=1, j\neq i}^{n} \left(\frac{x - x_j}{x_i - x_j} \right) \quad \forall i = 1, ..., n.$$

Remarque : Les polynômes de Lagrange s'identifient au symbole de Kronecker δ_{ij} pour tout $i = 1, ..., n$ i.e

$$l_i(x_k) = \prod_{j=1, j\neq i}^{n} \left(\frac{x_k - x_j}{x_i - x_j} \right) = \begin{cases} 0 & \text{si } k \neq i \\ 1 & \text{si } k = i \end{cases}$$

Comment peut-on déterminer les polynômes de Lagrange ?

Nous présentons une méthode pratique pour la détermination de ces polynômes. En considérant le tableau suivant :

$$\begin{pmatrix} x-x_1 & x_1-x_2 & x_1-x_3 & \cdots & x_1-x_n \\ x_2-x_1 & x-x_2 & x_2-x_3 & \cdots & x_2-x_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n-x_1 & x_n-x_2 & x_n-x_3 & \cdots & x-x_n \end{pmatrix}$$

Alors, les polynômes de Lagrange sont le rapport du produit des termes diagonaux du tableau sur le produit des termes de la $(i+1)^{\text{éme}}$ ligne du tableau.

Exemple

Quels sont les trois polynômes de Lagrange associés aux valeurs suivantes :$x_1 = 0$,$x_2 = 4$,$x_3 = -4$ et $x_4 = 2$?

Solution : On a le tableau suivant pour $n = 4$:

$$\begin{pmatrix} x-x_1 & x_1-x_2 & x_1-x_3 & x_1-x_4 \\ x_2-x_1 & x-x_2 & x_2-x_3 & x_2-x_4 \\ x_3-x_1 & x_3-x_2 & x-x_3 & x_3-x_4 \\ x_4-x_1 & x_4-x_2 & x_4-x_3 & x-x_4 \end{pmatrix} = \begin{pmatrix} x & -4 & 4 & -2 \\ 4 & x-4 & 8 & 2 \\ -4 & -8 & x+4 & -6 \\ 2 & -2 & 6 & x-2 \end{pmatrix}$$

Ainsi,

$$l_1(x) = \frac{\text{produit des termes diagonaux}}{\text{produit des termes de la } 2^{\text{éme}} \text{ ligne}} = \frac{x(x-4)(x+4)(x-2)}{(4)(x-4)(8)(2)} = \frac{x(x+4)(x-2)}{(4)(8)(2)}$$

$$l_2(x) = \frac{\text{produit des termes diagonaux}}{\text{produit des termes de la } 3^{\text{éme}} \text{ ligne}} = \frac{x(x-4)(x+4)(x-2)}{(-4)(-8)(x+4)(-6)} = \frac{x(x-4)(x-2)}{(-4)(-8)(-6)}$$

$$l_3(x) = \frac{\text{produit des termes diagonaux}}{\text{produit des termes de la } 4^{\text{éme}} \text{ ligne}} = \frac{x(x-4)(x+4)(x-2)}{(2)(-2)(6)(x-2)} = \frac{x(x-4)(x+4)}{(2)(-2)(6)}$$

Définition 2.1.2. On appelle polynôme d'interpolation de Lagrange aux points $x_i = 1, ..., n$, tout polynôme défini sous la forme suivante :

$$L(x) = \sum_{i=1}^{n} f(x_i) l_i(x) = \sum_{i=1}^{n} l_i(x) y_i.$$

Ce polynôme est l'unique polynôme de degré $n-1$ vérifiant $L(x_i) = f(x_i) = y_i$ pour tout $i = 1, ..., n$.

Exemples d'application

- **Cas où la fonction f est inconnue :**
  Construire le polynôme d'interpolation de Lagrange de la fonction f aux points $x_1 = 0$ avec $f(x_1) = -1$ et $x_2 = 1$ avec $f(x_2) = 2$.

  Solution :
  Nous avons :
  $$L(x) = \sum_{i=1}^{2} f(x_i) l_i(x) = -l_1(x) + 2l_2(x)$$
  Nous savons que :
  $$l_1(x) = \prod_{j=1, j\neq 1}^{2} \left(\frac{x - x_j}{x_1 - x_j}\right) = \left(\frac{x - x_2}{x_1 - x_2}\right) = \frac{x-1}{0-1} = -x + 1.$$
  $$l_2(x) = \prod_{j=1, j\neq 2}^{2} \left(\frac{x - x_j}{x_2 - x_j}\right) = \left(\frac{x - x_1}{x_2 - x_1}\right) = \frac{x-0}{1-0} = x.$$
  Par conséquent,
  $$L(x) = -l_1(x) + 2l_2(x) = -(-x + 1) + 2x = 3x - 1.$$

- **Cas où la fonction f est connue :**
  Soit f une fonction définie par :
  $$f(x) = \sin(\pi x).$$
  Écrire le polynôme d'interpolation de Lagrange de la fonction f aux points $x_1 = 0$, $x_2 = \frac{1}{6}$ et $x_3 = \frac{1}{2}$.
  Solution :
  1. Calculons les fonctions $f(x_i)$ pour tout $i = 1, 2, 3$:
     $f(x_1) = f(0) = 0$,$f(x_2) = f(\frac{1}{6}) = \frac{1}{2}$,$f(x_3) = f(\frac{1}{2}) = 1$.
  2. Calculons le polynôme d'interpolation de Lagrange $L(x)$ pour tout $i = 1, 2, 3$:
  $$L(x_) = \sum_{j=1, j\neq i}^{3} f(x_i) l(x_i) = 0\left(\frac{(x-\frac{1}{6})(x-\frac{1}{2})}{(-\frac{1}{6})(-\frac{1}{2})}\right) + \frac{1}{2}\left(\frac{x(x-\frac{1}{2})}{\frac{1}{6}(\frac{1}{6}-\frac{1}{2})}\right) + 1\left(\frac{x(x-\frac{1}{6})}{\frac{1}{2}(\frac{1}{2}-\frac{1}{6})}\right)$$
  $$= -3x^2 + \frac{7}{2}x.$$

Exemple d'application sous langage R

Dans l'exemple suivant, nous voulons interpoler un polynôme de Lagrange de degré 6 entre les points $(0, 10), (1, 5), (2, 2)$et$(3, 1)$:

```
x <- seq(0, 3, length = 100)
f= 10 / (1 + x^2)
plot(x,f, type = "l", xlim = c(0, 3), ylim = c(0, 10), ylab = "",
    col = "black")

par(new = TRUE)
plot(x, 10 - 6.4 * x^2 + 1.5 * x^4 - 0.1 * x^6, type = "l", xlim =
    c(0, 3), ylim = c(0, 10), ylab = "", lty = 2, col = "blue")
abline(v = 1)
abline(v = 2)

par(new = TRUE)
legend(1.8, 9, c("courbe", "Lagrange"), col = c("black", "blue"),
    lty = c(1, 2, 1))
```

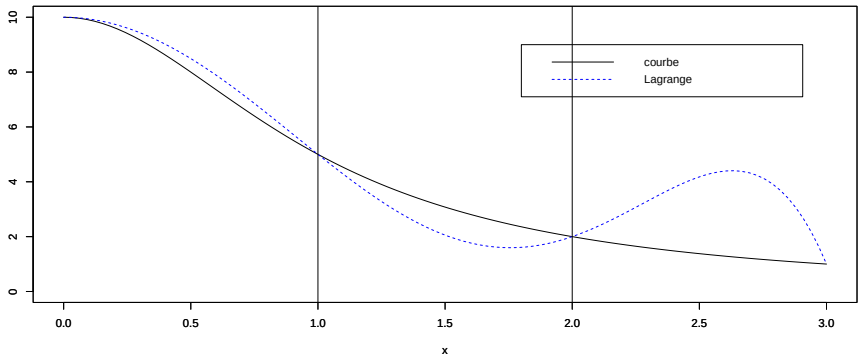

FIGURE 2.7 – Interpolation d'un polynôme de Lagrange

Interprétation : On observe que même si la courbe de Lagrange est précise aux points de données, on remarque qu'il existe des écarts importants entre ces points, ce qui rend l'approximation moins fiable.

2.1.2 Polynôme d'interpolation d'Hermite

Principe de l'interpolation d'Hermite

Soient $x_0, x_1, ..., x_n, (n+1)$ points deux à deux distincts de l'intervalle $[a, b]$ et soit $f \in \mathcal{C}^1([a, b])$ une fonction dont on connaît les valeurs $f(x_i)$ et $f^{'}(x_i)$ pour tout $i = 1, .., n$.
Nous cherchons à trouver un polynôme $P(x)$ qui interpole à la fois f et f' aux points x_i pour tout $i = 1, .., n$. Ainsi, selon la definition d'un polynôme d'interpolation $P(x)$ doit satisfaire simultanément les deux conditions suivantes :

$$P(x_i) = f(x_i) \quad \forall i = 1, .., n. \tag{2.13}$$

$$P^{'}(x_i) = f^{'}(x_i) \quad \forall i = 1, ..., n. \tag{2.14}$$

Définition 2.1.3. Le polynôme d'interpolation d'Hermite de degré $(2n+1)$est défini comme suit :

$$P_{2n+1}(x) = \sum_{i=1}^{n} f(x_i)H_i(x) + \sum_{i=1}^{n} f^{'}(x_i)V_i(x).$$

Où :

- Les $H_i(x)$ sont connus sous le nom de *"polynômes de base d'Hermite"* tels que :

$$H_i(x) = \left[1 - 2(x - x_i)L_i^{'}(x_i)\right] L_i^2(x).$$

- Les $V_i(x)$ sont les polynômes de base d'Hermite dérivés tels que :

$$V_i(x) = (x - x_i)L_i^2(x).$$

Où :

- Les $L_i(x)$ sont des polynômes de Lagrange tels que :

$$L_i(x) = \prod_{j=1, j\neq i}^{n} \frac{(x - x_j)}{(x_i - x_j)}.$$

- Les $L_i^{'}(x_i)$ sont les déviées des polynômes de Lagrange tels que :

$$L_i^{'}(x_i) = \sum_{j=1, j\neq i}^{n} \frac{1}{(x_i - x_j)}.$$

Exemple d'application

Soit $f \in C^2([0, 1])$, déterminer le polynôme d'interpolation d'Hermite $P_3(x)$ de degré ≤ 3 tel que :

$$P_3(0) = f(0) \quad , \quad P_3(1) = f(1).$$
$$P_3'(0) = f'(0) \quad , \quad P_3'(1) = f'(1).$$

Solution

1. Calculons les polynômes de Lagrange $L_i(x)$ et leurs dérivées $L_i^{'}(x)$ aux points $x_1 = 0$ et $x_2 = 1$:

$$L_1(x) = 1 - x \quad , \quad L_2(x) = x.$$
$$L_1^{'}(x) = -1 \quad , \quad L_2^{'}(x) = 1.$$

2. Calculons $H_i(x)$ et $V_i(x)$ aux points $x_1 = 0$ et $x_2 = 1$:

$$H_1(x) = \left[1 - 2(x - x_1)L_1^{'}(x_1)\right] L_1^2(x) = (1 + 2x)(1 - x)^2.$$

$$H_2(x) = \left[1 - 2(x - x_2)L_2^{'}(x_2)\right] L_2^2(x) = (3 - 2x)x^2.$$

$$V_1(x) = (x - x_1)L_1^2(x) = x(1 - x)^2.$$
$$V_2(x) = (x - x_2)L_2^2(x) = (x - 1)x^2.$$

3. Le polynôme d'interpolation d'Hermite $P_3(x)$ aux points $x_1 = 0$ et $x_2 = 1$ est donné par :

$$\begin{aligned} P_3(x) &= \sum_{i=1}^{2} f(x_i)H_i(x) + \sum_{i=1}^{2} f^{'}(x_i)V_i(x) \\ &= f(x_1)H_1(x) + f(x_2)H_1(x) + f^{'}(x_1)V_1(x) + f^{'}(x_2)V_2(x) \\ &= f(0)(1+2x)(1-x)^2 + f(1)(1-2x)x^2 + f'(0)x(1-x)^2 + f'(1)(x-1)x^2 \\ &= [2f(0) - 2f(1) + f'(0) + f'(1)]\, x^3 + [-3f(0) + 3f(1) - 2f'(0) - f'(1)]\, x^2 \\ &\quad + f'(0)x + f(0). \end{aligned}$$

Exemple sous langage R

Exemple 2.1.1. L'exemple suivant montre comment utiliser l'interpolation d'Hermite pour évaluer un polynôme en un point spécifique.

Entrées :

- xi : Un vecteur contenant les abscisses xi.
- fxi : Un vecteur contenant les valeurs $f(xi)$ de la fonction à interpoler pour les valeurs xi correspondantes.
- $fpxi$: Un vecteur contenant les valeurs $f'(xi)$.
- x : Un point auquel évaluer le polynôme d'interpolation d'Hermite.

Sortie : La valeur du polynôme d'interpolation d'Hermite en x.

```
polynomial_hermite <- function(xi, fxi, fpxi, x) {
  n <- length(xi)
  for (i in 1:n) {
    if (x == xi[i]) {
      return(fxi[i])
    }
  }

  p <- 0
  L <- rep(0, n)
  c <- rep(0, n)

  for (i in 1:n) {
    L[i] <- 1
    for (j in 1:n) {
      if (i != j) {
        L[i] <- L[i] * (x - xi[j]) / (xi[i] - xi[j]) # Calculation
            of L_i(x).
        c[i] <- c[i] + 1 / (xi[i] - xi[j]) # Calculation of L'_i(x_
            i).
      }
    }
    p <- p + ((1 - 2 * (x - xi[i]) * c[i]) * fxi[i] + (x - xi[i]) *
        fpxi[i]) * L[i]^2 # Evaluation of p_{2n+1}(x).
  }
  return(p)
}

```

```r
# Example usage:
xi <- c(-1, 0, 1, 2, 3)
fxi <- c(-2, -1, 0, 3, 2)
fpxi <- c(1, 2, 1, 4, -2)
p <- polynomial_hermite(xi, fxi, fpxi, 2.5)
print(p)
[1] 3.924978  #the result of evaluating the Hermite interpolation
    polynomial at x = 2.5
```

Théorème 3. ***[25]***
Il existe un unique polynôme d'interpolation $P(x)$, de degré inférieur ou égal à $(2n+1)$ de la fonction f aux points x_i pour tout $i = 1, ..., n$, vérifiant les équations (13) et (14) si et seulement si, tous les x_i sont distincts deux à deux .

Remarque : Dans l'interpolation polynomiale traditionnelle le degré du polynôme augmente avec le nombre de points à interpoler. Par conséquent, cela conduit à un nombre élevé d'opérations et à une complexité de calcul importante.
Pour pallier ce problème, les chercheurs ont proposé l'utilisation de splines interpolantes. Cette méthode se base sur la minimisation d'un critère fonctionnel qui définit la forme de la spline, indépendamment du nombre de points à interpoler.

Alors, quelle est la définition exacte de l'interpolation spline ?

2.2 Interpolation par morceaux

2.2.1 Définition

L'interpolation par morceaux (ou interpolation par spline) est une méthode d'interpolation utilisée pour construire une fonction f continue à partir d'un ensemble de points $(x_1, y_1), ..., (x_n, y_n)$. Elle divise l'intervalle entre les points en segments plus petits et utilise des polynômes de faible degré pour interpoler chaque segment. Ces polynômes sont choisis de manière à assurer la continuité de la courbe interpolée ainsi que la satisfaction des conditions d'interpolation ($f(x_i) = y_i$ pour tout $i = 1, \ldots, n$).

2.2.2 Exemples d'interpolation par spline

Interpolation par spline cubique

Définition 2.2.1. L'interpolation par spline cubique est une méthode robuste qui utilise des polynômes cubiques pour ajuster une courbe lisse à travers des données, en garantissant la continuité de la courbe interpolée, ainsi que celle de sa première et de sa seconde dérivée sans nécessiter de valeurs de dérivées premières supplémentaires.

Exemple d'application sous langage R

Dans cet exemple, nous avons comparé deux méthodes d'interpolation :

```r
# Define the x and y points for interpolation
x_points <- c(0, 1, 2, 3)
y_points <- c(10, 5, 2, 1)

```

```r
# Define the Lagrange interpolation function
lagrange <- function(x, y, xout) {
  n <- length(x)
  yout <- rep(0, length(xout))
  for (i in 1:n) {
    L <- 1
    for (j in 1:n) {
      if (i != j) {
        L <- L * (xout - x[j]) / (x[i] - x[j])
      }
    }
    yout <- yout + y[i] * L
  }
  return(yout)
}

# Calculate the Lagrange interpolation values
lagrange_vals <- lagrange(x_points, y_points, x)

# Plot the Lagrange interpolation with a blue dashed line
lines(x, lagrange_vals, type = "l", col = "blue", lty = 2)
abline(v = 1)
abline(v = 2)

# Define the piecewise polynomial functions
f1 <- (5 - 5.65385 * (x - 1) + 3.69231 * (x - 1)^2 + 4.34615 * (x -
    1)^3) * (0 <= x) * (x < 1)
f2 <- (2 - 1.38462 * (x - 2) + 0.57692 * (x - 2)^2 - 1.03846 * (x -
    2)^3) * (1 <= x) * (x < 2)
f3 <- (1 - 0.80769 * (x - 3) - 0.19231 * (x - 3)^3) * (2 <= x) * (x
    < 3)

# Plot the piecewise polynomial functions with a black line
lines(x, f1 + f2 + f3, type = "l", col = "black")

# Add a legend to the plot
legend(1.8, 9, c("courbe", "Lagrange", "spline"), col = c("red", "
   blue", "black"), lty = c(1, 2, 1))
```

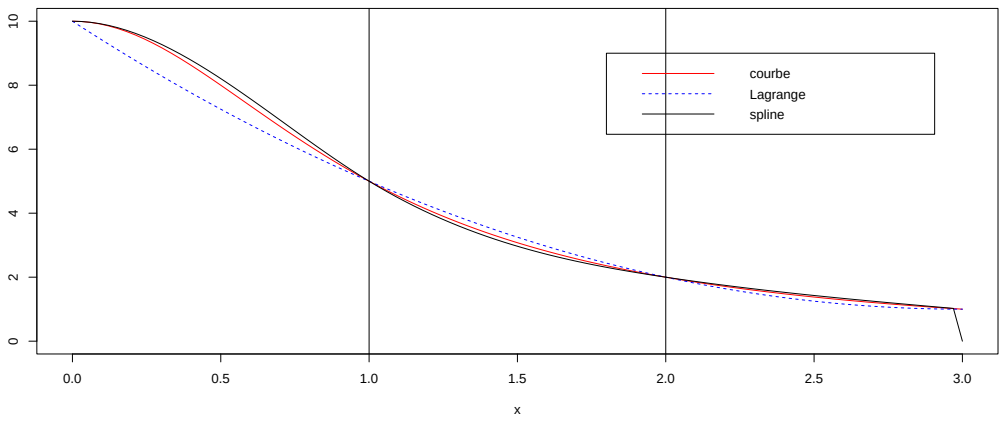

FIGURE 2.8 – Comparaison de deux méthodes d'interpolation

Interprétation du graphique : Ce graphe représente une comparaison de deux méthodes d'interpolation : interpolation de Lagrange et l'interpolation spline où la courbe de l'interpolation de Lagrange suit les points de données sauf entre 0 et 1, on observe des ballant importants. Tandis que l'interpolation par spline offre une courbe lisse et plus proche de la courbe référence.
Alors, l'interpolation par spline est préférable à l'interpolation de Lagrange, car elle fournit une approximation fiable et précise pour la courbe de référence.

Interpolation par spline d'Hermite

Définition 2.2.2. L'interpolation par spline d'Hermite permet de construire une fonction interpolante continue et dont la première dérivée est également continue, en utilisant des polynôme d'Hermite cubiques qui satisfont à la fois les conditions de l'interpolation et les conditions de la continuité des dérivées premières.

Exemple sous langage R

Dans cet exemple, nous comparons la méthode de l'interpolation par spline cubique avec d'interpolation par spline d'Hermite :

```
# Example data
x <- c(0, 1, 2, 3, 4, 5)
y <- c(1, 2, 1, 0, 2, 3)

# Cubic spline interpolation
spline_interp <- spline(x, y, n = 100)
plot(x, y, type = "l", main = "Comparison␣of␣Interpolations", xlab
    = "x", ylab = "y")
lines(spline_interp$x, spline_interp$y, col = "blue", lwd = 2)

# Hermite spline interpolation using 'pracma'
library(pracma)
hermite_interp <- pchip(x, y, seq(min(x), max(x), length.out = 100)
    )

# Add Hermite interpolated lines
lines(seq(min(x), max(x), length.out = 100), hermite_interp, col =
    "red", lwd = 2)

# Legend
legend("topright", legend = c("Cubic-Spline", "Hermite-Spline"),
    col = c("blue", "red"), lwd = 2)
```

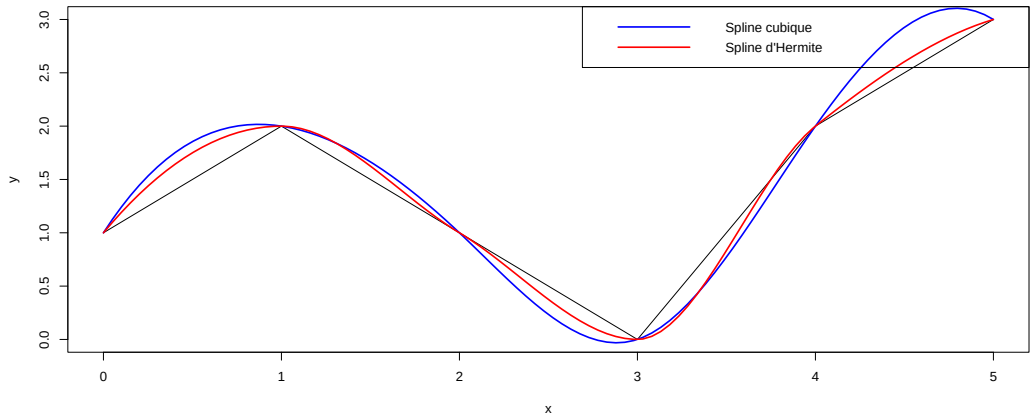

FIGURE 2.9 – Comparaison de deux méthodes d'interpolation par morceaux

Interprétation du graphique : Dans ce graphe, on voit que la spline cubique crée une courbe plus lisse entre les points, tandis que la spline d'Hermite suit mieux les changements des données. La courbe rouge est plus précise lorsque les données changent de manière significative, alors que la courbe bleue est plus adaptée si on veut une transition plus douce entre les points.

Interpolation par des B-splines

Définition 2.2.3. L'interpolation par des B-splines consiste à ajuster une courbe lisse à travers un ensemble de points, chaque segment entre ces points est interpolé en utilisant une combinaison linéaire de fonctions de base définies localement sur ce segment.

Exemple sous langage R

Dans cet exemple, nous avons réalisé une interpolation par B-splines sur le jeu de données "iris" afin de visualiser la relation entre la longueur et la largeur des pétales des iris.

```r
library(splines)
# Load the iris dataset
data(iris)

# Select a variable to use
x <- iris$Petal.Length
y <- iris$Petal.Width

# Number of knots for B-splines
k <- 5

# Interpolation by B-splines
bspline_fit <- lm(y ~ ns(x, df = k))

# Plot the interpolation
plot(x, y, pch = 16, col = "blue", xlab = "Petal-Length", ylab = "
    Petal-Width")
```

```
lines(sort(x), predict(bspline_fit)[order(x)], col = "red", lwd =
    2)
legend("topleft", legend = c("Data", "B-spline"), col = c("blue", "
    red"), pch = c(16, NA), lwd = 2)
```

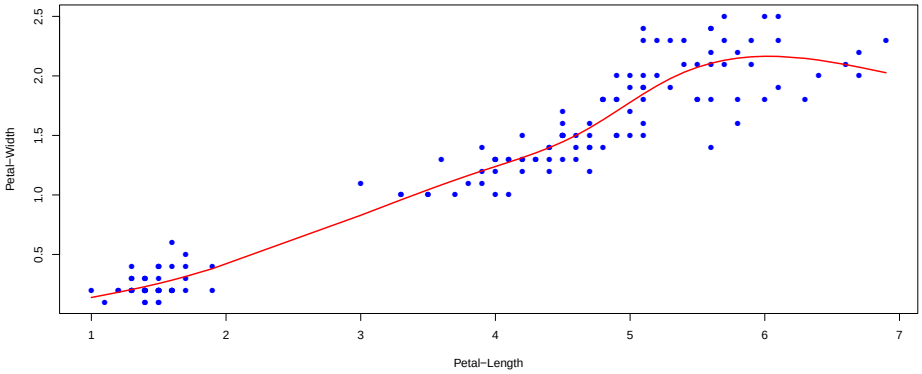

FIGURE 2.10 – Interpolation par B-splines

Interprétation du graphique : Le graphe montre un nuage de points qui représente la longueur et la largeur des pétales, où les points bleus représentent des mesures de la longueur et la largeur des pétales et la ligne rouge montre une relation non linéaire entre ces variables.
On remarque que, les deux variables augmentent de manière plus ou moins linéaire. Après un certain points, le taux d'augmentation de la largeur des pétales ralentit. Cependant, à partir des longueurs supérieur à 6 la largeur commence à diminuer. Cela confirme que la relation entre les variables est non linéaire et les b-splines est la seule méthode qui permet d'adapter à ce changement de tendance observée dans les données.

2.2.3 Espace de Sobolev

Définition 2.2 : Soient Ω un ouvert de $\mathbb{R}^n$ et $1 \leq p < \infty$.On définit l'espace de Sobolev $W_p^r(\Omega)$ comme suit :

$$W_p^r(\Omega) = \{f \in L^p(\Omega), f^{(k)} \in L^p(\Omega), \text{pour tout} |k| < r\}$$

En particulier, si $\Omega = [a, b]$ et $p = 2$ on aura :

$$W_2^r[a,b] = \{f : [a,b] \to \mathbb{R}; f^{(k)} \text{absolument continue pour} \quad k = 0, ..., r-1 \text{et} \int_a^b (f^{(r)}(x))^2 dx < \infty\}.$$

Remarque : On dit que f est une fonction absolument continue, s'il existe un réel a et une fonction g integrable tels que :

$$f(x) = \int_a^x g(t)dt.$$

Dans cette section, nous présentons le problème d'interpolation par une fonction lisse(spline) qui appartienne à l'espace de Sobolev W_2^2.

2.2.4 Existence et unicité des splines d'interpolation

Théorème 4. ***[2]***

Étant donnés n points (x_i, y_i) d'abscisses distinctes dans l'intervalles $[a, b]$ et $n \geq r$. Il existe une fonction et une seule f de l'espace de Sobolev $W_2^r[a, b]$ telle que :

(i). f satisfait les conditions d'interpolation

$$f(x_i) = y_i, i = 1, .., n.$$

(ii). f minimise la quantité $\int_a^b f^{(r)}(x)^2 dx$ dans l'ensemble des fonctions $W_2^r[a, b]$ qui satisfait les conditions d'interpolation.

De plus, cette fonction est une spline polynomiale naturelle d'ordre $(2r)$ ayant des noeuds aux positions $x_1, \ldots, x_n$.

Le théorème suivant démontre que l'interpolation par spline cubique est la seule méthode qui minimise $\int_a^b (f''(x))^2 dx$ par rapport à toutes les fonctions dans $W_2^2[a, b]$.

Théorème 5. ***[2]***
Supposons que $n \geq 2$ et $\widehat{m}_\lambda$ est la spline cubique pour les valeurs $y_1, ..., y_n$ aux points $x_1, ..., x_n$, où $a < x_1 < .... < x_n < b$. Soit $\widetilde{m}$ une fonction dans $W_2^2[a, b]$ telle que $\widetilde{m}(x_i) = y_i, i = 1, .., n$. Alors,

$$\int_a^b \widehat{m}''(x)^2 dx \geq \int_a^b (\widetilde{m}''(x))^2 dx.$$

On a l'égalité si et seulement si $\widetilde{m}$ et $\widehat{m}$ sont identiques.

III. Régression spline

Nous rappelons que l'interpolation vise à trouver une fonction qui passe exactement par les points à interpoler, tandis que la régression cherche à ajuster une fonction simple à ces points. Dans cette approche n'est plus de passer exactement par les points, mais de s'en approcher et d'obtenir une fonction simple m qui les représentent. Cette problématique est d'autant plus crucial dans la pratique, car les observations des évaluations sont souvent perturbées par du bruit, résultent par exemple d'imprécisions de mesure.

Pour résoudre cette dernière, de nombreux auteurs recommandent l'utilisation de polynômes splines. Les splines de lissage [1]trouvent leurs origines dans les travaux de Whittaker(1923) et ont été développées par Schoenberg(1964) et Reinsch.

Supposons que $\mathbb{I} = [a; b]$ un intervalle de $\mathbb{R}$ tel que $a < x_1 < \ldots < x_n < b$, et considérons le modèle non paramétrique suivant :

$$y_i = m(x_i) + \epsilon_i, \quad \forall i = 1, ...n.$$

1. Sont des fonctions par morceaux, définies par des polynômes, qui équilibrent l'ajustement aux données et la régularité via un paramètre de lissage.

Où m est une fonction lisse [2] inconnue dans $W_2^2[a,b]$, y_i sont des valeurs observées de la variable réponse Y, x_i sont des valeurs observées de la variable X et ϵ_i sont des erreurs gaussiennes de moyenne 0 et de variance σ^2. L'objectif principal est d'approximer la fonction inconnue m, en cherchant à trouver un équilibre entre un bon ajustement aux données et une estimation lisse.

En combinant ces deux critères, l'estimateur optimal est la fonction $\widehat{m_\lambda}$ qui minimise

$$\sum_1^n (y_i - m(x_i))^2 + \lambda \int_a^b (m^{(r)}(x))^2 dx, \lambda > 0,$$

par rapport à toutes les fonctions m dans $W_2^r[a,b]$.

L'estimateur $\hat{m}_\lambda$ est appelé la spline de lissage. Le paramètre λ appelé paramètre de lissage, appartient à l'intervalle $[0, +\infty[$ et contrôle le compromis entre le lissage et le bon ajustement. Lorsque le paramètre λ est grand, on accorde plus d'importance au lissage, ce qui signifie que l'estimateur est plus lisse. Inversement, quand les valeurs de λ sont proches de 0, l'importance est mise sur la qualité de l'ajustement, ce qui donne un estimateur flexible.

2.1 Spline de lissage

Les splines de lissage déterminent une estimation en minimisant un critère combinant la qualité de l'ajustement, mesurée par la somme des résidus au carré et la qualité de lissage.

Supposons que $x_1 < x_2 < .... < x_n$, l'estimation de la fonction m par les splines de lissage résout un problème de minimisation. Visant à trouver la fonction $\hat{m}_\lambda$ qui minimise la somme des carrés des résidus pénalisée, également appelée erreur quadratique pénalisée.

$$S(m) = \sum_{i=1}^n (y_i - m(x_i))^2 + \lambda \int_a^b (m^{(r)}(x))^2 dx$$

où $m^{(r)}$ est la dérivée d'ordre r de la fonction m. Autrement dit :

$$\hat{m}_\lambda(x) = argmin \quad \sum_{i=1}^n (y_i - m(x_i))^2 + \lambda \int_a^b (m^{(r)}(x))^2 dx.$$

Théorème 6. ***[2]***
Le minimum de problème $S(m) = \sum_{i=1}^n (y_i - m(x_i))^2 + \lambda \int_a^b (m^{(r)}(x))^2 dx$ *admet une solution unique* $\hat{m}_\lambda$ *qui est une fonction spline dans l'ensemble* $S_{2k}(x_1 \dots x_n)$.

Pour $r = 2$, nous cherchons à trouver la fonction m qui minimise

$$S(m) = \sum_{i=1}^n (y_i - m(x_i))^2 + \lambda \int_a^b (m''(x))^2 dx,$$

2. Est une fonction qui est continue et possède des dérivées continues, sans variations abruptes ou discontinuités.

par rapport à toutes les fonctions m dans $W_2^2[a,b]$.

On peut réécrire $S(m)$ sous la forme suivante :

$$S(m) = (Y-M)^t(Y-M) + \lambda M^t A M$$

où $Y=(y_1,...,y_n)^t$,$A = QR^{-1}Q^t$ et $M=(M_1,...,M_n)$ tel que : $M_i = \hat{m}_\lambda(x_i)$.

2.1.1 Existence et unicité de la spline de lissage minimisante

Théorème 7. ***[2]***
Soient (x_1,y_1),.....,(x_n,y_n), n points donnés dans l'intervalle $[a,b]$ et un réel $\lambda > 0$. Il existe une fonction et une seule $\hat{m}_\lambda$ de l'espace de Sobolev$W_2^2[a,b]$ qui minimise la quantité

$$S(m) = \sum_{i=1}^{n}(y_i - m(x_i))^2 + \lambda \int_a^b (m''(x))^2 dx,$$

par rapport à toutes les fonctions m dans $W_2^r[a,b]$.

Preuve :
On a :

$$\begin{aligned} S(m) &= (Y-M)^t(Y-M) + \lambda M^t A M \\ &= (Y^t - M^t)(Y-M) + \lambda M^t A M \\ &= Y^tY - Y^tM - M^tY + M^tM + \lambda M^t A M \\ &= Y^tY - 2Y^tM + M^t(I+\lambda A)M. \end{aligned}$$

Dérivons $S(m)$ par rapport a M :

$$\begin{aligned} \frac{dS(m)}{dM} &= \frac{d}{dM}(Y^TY - 2Y^TM + \lambda M^T(I+A)M) \\ &= -2Y + 2(I+\lambda A)M \end{aligned}$$

$$\frac{dS(m)}{dM} = 0 \Leftrightarrow M = (I+\lambda A)^{-1}Y.$$

Sa dérivée seconde est donnée par :

$$\frac{d^2S(m)}{dM^2} = 2(I+\lambda A).$$

Comme λK est non négative, alors $(I+\lambda A)$ est strictement définie positive.

Donc : $\hat{M} = (I+\lambda A)^{-1}Y$ est bien un minimum.

Théorème 8. ***[2]***
Supposons que $n \geq 0$ et que le paramètre de lissage $\lambda > 0$, alors $\widehat{m_\lambda}$est une spline cubique telle que :

$$M = (I+\lambda A)^{-1}Y$$

et pour toute m dans $W_2^2[a,b]$, on a :

$$S(\widehat{m_\lambda}) \leq S(m).$$

2.1.2 Propriétés de l'estimateur splines de lissage

Soit $\hat{m}_\lambda$ l'estimateur spline, alors $\hat{m}_\lambda$ est définit comme suit :

$$\hat{m}_\lambda = (I + \lambda A)^{-1} Y = P_\lambda Y.$$

avec P_λ est appelée matrice chapeau ou matrice de lissage.

1. Le biais de $\hat{m}_\lambda$ est défini par :

$$B(\hat{m}_\lambda, m) = \mathbb{E}(\hat{m}_\lambda - m) = (P_\lambda - I)m.$$

2. La variance de $\hat{m}_\lambda$ est définie comme suit :

$$Var(\hat{m}_\lambda) = \mathbb{E}\left[\hat{m}_\lambda - \mathbb{E}(\hat{m}_\lambda)\right]^2 = \sigma^2 tr(P_\lambda^t).$$

2.1.3 Exemple d'application sous langage R

Dans cet exemple, on crée un jeu de données avec deux variables x et y et un nuage de point pour visualiser la relation entre elles.

- **Création de jeu de données :**

```
# Create the data frame
df <- data.frame(x = 1:20,
                 y = c(2, 4, 7, 9, 13, 15, 19, 16, 13, 10,
                       11, 14, 15, 15, 16, 15, 17, 19, 18, 20))
# Create a scatterplot
plot(df$x, df$y, cex = 1.5, pch = 19)
```

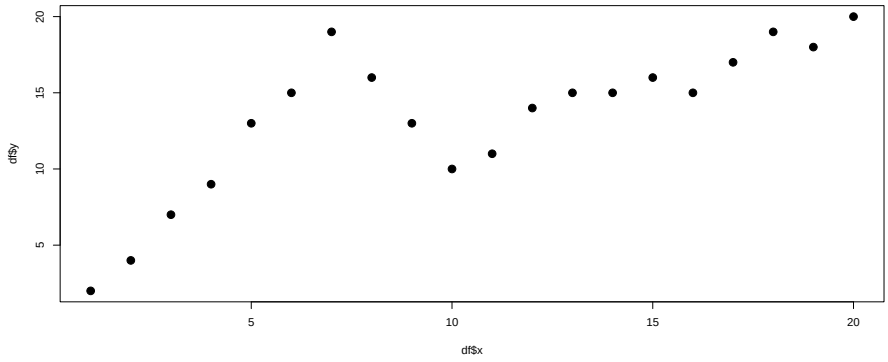

FIGURE 2.11 – Représentation graphique de points.

Interprétation du graphique : À partir de cette figure, on voit que la relation entre x et y n'est pas linéaire et on observe des changements de comportement des données à x=7 et x = 10, ce qui indique que ces points sont des nœuds.

- **Application de la régression linéaire :**
En utilisant la fonction $lm()$ pour ajuster un modèle de régression linéaire simple à ce jeu de données.

```r
# Fit a simple linear regression model
linear_fit <- lm(df$y ~ df$x)

# Display the summary of the model
summary(linear_fit)
```

```
Call:
lm(formula = df$y ~ df$x)

Residuals:
    Min      1Q  Median      3Q     Max
-5.2143 -1.6327 -0.3534  0.6117  7.8789

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.5632     1.4643   4.482 0.000288 ***
df$x          0.6511     0.1222   5.327  4.6e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.152 on 18 degrees of freedom
Multiple R-squared:  0.6118,    Adjusted R-squared:  0.5903
F-statistic: 28.37 on 1 and 18 DF,  p-value: 4.603e-05
```

Interprétation : L'analyse de régression linéaire montre que à chaque fois que la variable indépendante augmente d'une unité, la variable dépendante augmente en moyenne de 0.6511 unité. En d'autres termes, il existe une relation positive entre les variables.

Le R^2_{adj} de 0.5903 indique que 59.03% de la variabilité de y est expliquée par x et la statistique F de 28.37 avec p-value inférieur à 0.05 confirme l'existence d'une relation linéaire entre ces variables.

Donc, le modèle est donné sous la forme suivante :

$$y = 6.5632 + 0.6511 * x$$

- **Visualisation de la régression linéaire simple :**

```r
# Create a scatterplot
plot(df$x, df$y, cex = 1.5, pch = 19)

# Add the regression line to the scatterplot
abline(linear_fit)
```

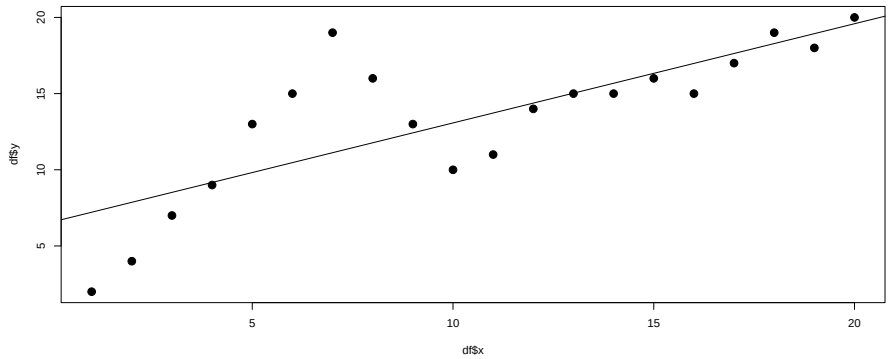

FIGURE 2.12 – Représentation graphique de régression linéaire.

Interprétation : À partir de ce graphique, on voit que la droite de la régression linéaire simple ne correspond pas bien aux données.

- **Application de la régression spline :**
On utilise la fonction *bs*() de package "splines" pou ajuster un modèles de régression spline avec deux nœuds $x = 7$ et $x = 10$.

```r
# Load the splines library
library(splines)

# Fit a spline regression model
spline_fit <- lm(df$y ~ bs(df$x, knots = c(7, 10)))

# Display the summary of the spline regression model
summary(spline_fit)
```

```
Call:
lm(formula = df$y ~ bs(df$x, knots = c(7, 10)))

Residuals:
     Min       1Q   Median       3Q      Max
-2.84883 -0.94928  0.08675  0.78069  2.61073

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                      2.073      1.451   1.429    0.175
bs(df$x, knots = c(7, 10))1      2.173      3.247   0.669    0.514
bs(df$x, knots = c(7, 10))2     19.737      2.205   8.949 3.63e-07 ***
bs(df$x, knots = c(7, 10))3      3.256      2.861   1.138    0.274
bs(df$x, knots = c(7, 10))4     19.157      2.690   7.121 5.16e-06 ***
bs(df$x, knots = c(7, 10))5     16.771      1.999   8.391 7.83e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.568 on 14 degrees of freedom
Multiple R-squared:  0.9253,    Adjusted R-squared:  0.8987
F-statistic:  34.7 on 5 and 14 DF,  p-value: 2.081e-07
```

Interprétation des résultats : À partir de ces résultats, on peut avoir que :

Les coefficients associées aux termes 2, 4 et 5 sont significativement différents de zéro avec des p-values très faibles (inférieur à 0.05), cela signifie que ces segments spécifiques de la variable indépendante ont une forte influence sur la variable dépendante.

Le coefficient de détermination ajusté R^2_{adj} de valeur 0.8987 indique que le modèle explique 89.87% de variation de la variable dépendante. De plus, le test de Frisher de p-value
2.081×10^{-7} confirme la significativité globale du modèle.

- **Visualisation de la régression spline :**

```r
# Calculate predictions using the spline regression model
x_lim <- range(df$x)
x_grid <- seq(x_lim[1], x_lim[2])
preds <- predict(spline_fit, newdata = list(x = x_grid))
# Create a scatterplot with the predictions from the spline
   regression
plot(df$x, df$y, cex = 1.5, pch = 19)
lines(x_grid, preds)
```

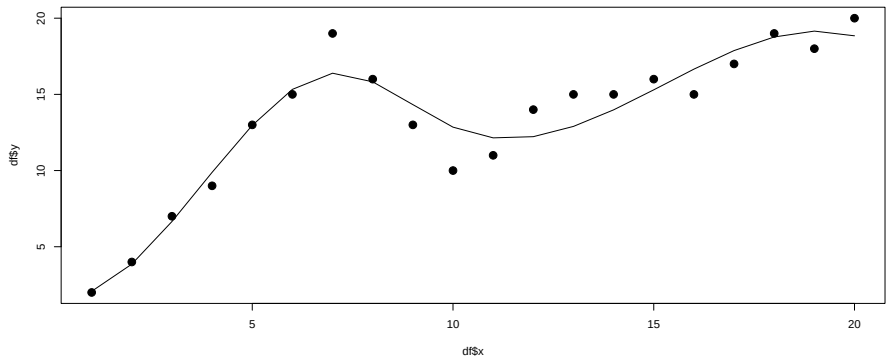

FIGURE 2.13 – Représentation graphique de régression spline.

Interprétation du graphique : À l'aide de ce graphique, on constate que le modèle de regression spline est le plus approprié pour les données, car il s'ajuste à la majorité d'entre elles.

- **Conclusion :** La valeur de R^2_{adj} du modèle de régression spline est plus élevé par rapport à celle du modèle linéaire simple, ce qui nous indique que le modèle de régression spline est le plus adapté aux données.

Remarque : En pratique, on détermine les nœuds en fonction des zones où les points montrent des changements significatifs dans le comportement des données.

2.1.4 Remarque

Perperoglou Aris et al [16], présente une analyse détaillée des techniques de modélisation par splines dans le langage R, en particulier pour l'analyse des données dans la recherche médicale. Avec l'avancement des méthodes théoriques et informatiques, les splines sont devenues un outil crucial en régression statistique, notamment pour capturer les relations non linéaires dans les variables continues.
L'article examine les packages de logiciel R les plus couramment utilisés pour les splines, mettant en évidence les défis associés à leur utilisation, comme le choix des paramètres et avoir une bonne compréhension pour obtenir des résultats fiables.

Chapitre 3

Application au machine learning

Le machine learning prend ses origines dans les premières recherches en informatiques et en intelligence artificielle au milieu du $20^{\text{ème}}$siècle.

En 1950, le chercheur **Alan Turing** pose la question de savoir si une machine peut "penser" et "apprendre" comme les êtres humains dans son célèbre article "Computing Machinery and Intelligence", introduisant l'idée de l'apprentissage automatique. Peu après, en 1957, **Frank Rosenblatt** conçoit le premier modèle d'apprentissage automatique qui est capable d'ajuster ses paramètres en fonction des données.

Le machine learning est défini de différentes manières. Selon le mathématicien américain Arthur Samuel en 1959, c'est une science qui consiste à amener les ordinateurs à apprendre sans être explicitement programmés. En 1998, l'américain Tom Mitchell propose une autre définition, déclarant qu'une machine apprend lorsque sa performance dans l'exécution d'une tâche s'améliore avec des nouvelles expériences. Ainsi, le machine learning est la capacité d'une machine à déterminer quel calcul réaliser pour résoudre un problème spécifique.

Dans le domaine du machine learning, les méthodes d'interpolations jouent un rôle crucial dans la modélisation et la prédiction des données. Parmi elles, les splines se distinguent par leur capacité à fournir des approximations à la fois flexibles et précises.

Dans ce chapitre, nous aborderons les concepts clés du machine learning et le rôle essentiel que jouent les statisticiens dans ce domaine. Et comment les outils statistiques, tels que les B-splines et les splines permettent de construire des modèles qui s'adaptent aux données complexes. De plus, nous présenterons une application de la régression spline, illustrant son utilité dans une situation concrète.

3.1 Présentation du machine learning

L'apprentissage automatique est le processus par lequel un algorithme évalue et améliore ses performances sans intervention humaine, en répétant son exécution sur des ensembles de données jusqu'à obtenir des résultats pertinents de manière autonome.

Exemple

Imaginons qu'une entreprise souhaite déterminer le montant total dépensé par un client ou une cliente à partir de ses factures. Dans ce cas, il suffit d'appliquer un algorithme classique, tel qu'une simple addition (un algorithme d'apprentissage n'est pas nécessaire dans ce cas).
Maintenant, supposons que nous voulions utiliser ces mêmes factures pour déterminer quels produits le client est le plus susceptible d'acheter dans un mois. Bien que cela soit probablement lié, nous n'avons pas toutes les informations nécessaires pour le faire. Cependant, si nous disposons de l'historique d'achat d'un grand nombre d'individus, il devient possible d'utiliser un algorithme de machine learning pour élaborer un modèle prédictif nous permettant de répondre à cette question.

3.1.1 Pourquoi utiliser le machine learning ?

Le Machine Learning peut être utilisé pour résoudre :

- Des problèmes pour lesquels nous n'avons pas de solution (comme dans l'exemple de la prédiction d'achats ci-dessus)
- Des problèmes pour lesquels nous savons résoudre mais nous ne pouvons pas formaliser en termes d'algorithmes (comme la connaissance d'images).
- Des problèmes pour lesquels nous avons une solution mais avec des méthodes beaucoup trop exigeantes en capacités informatiques (par exemple, la prédiction des interactions entre molécules de grande taille où les simulations sont très lourdes).

Le machine learning est donc utilisé lorsque les données sont relativement abondantes mais les connaissances sont insuffisamment développées. Par conséquent, le machine learning peut également faciliter l'apprentissage humain : les méthodes créées par des algorithmes d'apprentissage peuvent mettre en évidence l'importance relative de différentes informations ou la façon dont elles collaborent pour résoudre un problème spécifique. Dans l'exemple de la prédiction d'achats, la compréhension du modèle nous permet d'analyser quelles caractéristiques des achats précédents peuvent prédire ceux à venir. Cette utilisation du machine learning est très observée dans la recherche scientifique : par exemple, pour déterminer quels gènes sont impliqués dans le développement de certains types de tumeurs et comment ou bien pour identifier les régions d'une image cérébrale qui permettent de prédire un comportement, ou encore pour reconnaître des objets astronomiques particuliers à partir d'images de télescope.

3.1.2 Quels sont les éléments constitutifs du machine learning ?

Le machine learning s'appuie sur deux principes fondamentaux :

— D'un côté, les données, qui sont les exemples à partir desquels l'algorithme va apprendre. Il existe deux grands types de données : les données labélisées et les données non labélisées. Les données labélisées sont accompagnées d'un label y, qui identifie la décision à prendre pour un échantillon, tandis que les données non-labélisées ne le sont pas. Bien que ces dernières soient plus difficiles à exploiter, elles sont beaucoup plus accessibles.

— D'un autre côté, l'algorithme d'apprentissage représente la procédure appliquée à ces données pour générer un modèle.

On appelle entraînement le fait de faire fonctionner un algorithme d'apprentissage sur un jeu de données. Ces deux éléments sont également importants l'un que l'autre. D'une part, aucun algorithme d'apprentissage ne pourra produire un modèle de qualité à partir de données non pertinentes. D'autre part, un modèle entraîné avec un algorithme inapproprié, même sur des données pertinentes, ne sera pas de bonne qualité.

3.1.3 Les Types de machine learning

Il existe plusieurs types d'apprentissage mais nous nous intéressons sur les deux principaux en machine learning : l'apprentissage supervisé et l'apprentissage non-supervisé. Chacune de ces catégories possède des particularités distinctes et est utilisée dans des contextes spécifiques en fonction des objectifs d'apprentissage.

1. **Apprentissage supervisé :** On dit que l'apprentissage est supervisé lorsque le modèle est entraîné sur un ensemble de données labélisées. C'est-à-dire, notre échantillon des données se présente sous forme de couples entrée-sortie.
2. **Apprentissage non-supervisé :** Dans l'apprentissage non-supervisé, les échantillons sont non-labélisés. Cela signifie que le modèle est formé sur les données où les valeurs de sortie ne sont pas connues et donc, l'algorithme doit analyser l'ensemble des caractéristiques pour identifier les structures communes entre ces dernières afin de prédire l'objectif.

3.1.4 Processus de machine learning

Le processus de machine learning est constitué de plusieurs étapes clés, chacune étant cruciale pour la réussite de l'application des algorithmes de machine learning.

1. **Définition des données :** Cette étape consiste à sélectionner et à préparer un ensemble de données. Ces données seront utilisées pour apprendre à résoudre le problème pour lequel il a été développé.
2. **Extraction des données :** Cette étape implique la collecte des données provenant de différentes sources. Elle peut également employer des techniques d'échantillonnage pour obtenir un sous-ensemble représentatif, ce qui simplifie le processus d'analyse.
3. **Exploration des données :** Une fois les données collectées, elles doivent être préparées, organisées et nettoyées pour obtenir des prédictions de qualité. Cela implique le traitement des valeurs manquantes et l'organisation des données pour l'analyse.
4. **Partition aléatoire de l'échantillon :** Les données sont ensuite divisées de manière aléatoire en ensembles d'apprentissage, de validation et de test, qui sont utilisés pour estimer l'erreur de prédiction et sélectionner le modèle pertinent.
5. **Configuration du modèle :** Cette étape consiste à sélectionner les algorithmes de machine learning en fonction des besoins spécifiques tels que la régression ou la classification. Il s'agit d'estimer le modèle pour une valeur

donnée d'un paramètre de complexité et d'optimiser ce paramètre en fonction de la technique d'estimation de l'erreur retenue.

6. **Comparaison des modèles :** Après l'optimisation et l'ajustement des modèles, on utilise l'ensemble de données de test ou d'autres critères d'évaluation pour comparer les modèles. L'objectif est de choisir le modèle qui offre les meilleures performances de prédiction sur des données non observées.
7. **Validation croisée :** Si la taille de l'échantillon de test est trop petite à l'étape (4) pour fournir une estimation fiable de l'erreur de prédiction, une validation croisée peut être effectuée. Cette approche permet d'estimer l'erreur de prédiction de manière plus robuste en la moyennant sur plusieurs cas.
8. **Choix de la méthode retenue :** La méthode retenue est sélectionnée en fonction de ses capacités de prévision, de sa robustesse et éventuellement de l'interprétabillité du modèle obtenu.
9. **Ré-estimation du modèle :** Une fois la méthode choisie, le modèle est réestimé en utilisant l'ensemble complet de données. Cela inclut les paramètres et la complexité du modèles optimisés lors des étapes précédentes.
10. **Exploitation du modèle :** Enfin, le modèle optimisé est utilisé pour effectuer des analyses sur la base de données.

3.1.5 Les techniques de machine learning

Il existe plusieurs méthodes pour faire des prédictions ou des décisions basées sur les données. Parmi elles :

- **Classification :** C'est un processus de machine learning qui utilise des données labélisées pour prédire la classe ou la catégorie à laquelle appartient une nouvelle donnée non-labélisé. Cette méthode est cruciale dans diverses applications telles que la détection de spams et la prévision de maladies.
- **Réseaux de neurones :** Ce sont des modèles informatiques inspirés par le cerveau humain, développés pour détecter des relations complexes dans les données. Ils sont essentiels dans l'apprentissage profond(deep learning) et sont largement utilisés dans des différents domaines tels que la reconnaissance vocale et la vision par l'ordinateur.
- **Régression :** C'est une méthode d'apprentissage supervisé utilisée pour modéliser la relation entre une variable dépendante et une ou plusieurs variables indépendantes. Elle aide à prédire des valeurs telle que la température.

3.1.6 Rôle du statisticien dans le machine learning

La statistique et le machine learning sont des domaines interconnectés où la statistiques fournit les fondements théoriques nécessaires pour comprendre et interpréter les modèles de machine learning tandis que ce dernier propose des techniques et des algorithmes avancés pour traiter de grandes quantités de données (big data) et réaliser des prédictions précises.

Les statisticiens jouent un rôle crucial dans l'apprentissage, en appliquant des principes méthodes analytiques pour construire de prédictions robustes. Ils sont responsables de la transformation et de la préparation des données afin de les rendre

adaptées aux algorithmes de machine learning . Ils sélectionnent les modèles appropriés et utilisent des techniques de validation pour garantir que les résultats sont fiables, interprétable et alignés avec les objectifs spécifique des organisations.

3.2 Splines et B-splines dans machine learning

Dans cette section, nous nous intéressons au rôle des fonctions splines et B-splines dans la technique de l'apprentissage automatique, notamment dans la méthode des réseaux de neurones.

3.2.1 Splines dans les réseaux de neurones

Les splines ont été appliquées aux réseaux de neurones principalement de deux méthodes différentes. L'approche la plus directe consiste à les utiliser comme fonctions d'activation [1]. Par exemple, les splines cubiques peuvent remplacer les fonctions traditionnelles, telles que les sigmoïdes, dans les couches du réseau, permettant ainsi une transformation des entrées. Cette approche permet aux réseaux de neurones à mieux capturer les non-linéarités complexes présentes dans les données, ce qui peut améliorer les performances du modèle sur les tâches de prédiction et de classification.

Les deux articles suivants illustrent cette méthode en utilisant des splines sur des problèmes de régression, démontrant une amélioration notable des performances. Campolucci et al [5] ont montré que les splines de Catmull-Rom, avaient en moyenne une erreur d'un ordre de grandeur inférieure à celle des réseaux sigmoïdes. Scardapane et al [19] ont montré que les splines cubiques approximait les données de test avec en moyenne 8% moins d'erreur que les réseaux sigmoïdes avec la même architecture.

L'autre approche consiste à utiliser les B-splines comme il sera expliqué dans la section suivante.

3.2.2 B-splines dans les réseaux de neurones

Dans les réseaux de neurones, les B-splines peuvent être intégrés en tant que fonctions d'activation ou utilisées dans des couches intermédiaires pour transformer les entrées. Cela permet au réseau de capturer les non-linéarités de manière plus efficace que les fonctions d'activation classiques.

Cette méthode donne naissance aux réseaux de neurones B-spline(BSNN [2]), qui sont des réseaux peu profonds composés uniquement d'une couche d'entrée et d'une couche de sortie avec une opération intermédiaire.
Cette opération transforme les entrées $x \in \mathbb{R}^m$ en hypersurfaces à l'aide de fonctions spline multivariées. La sortie de réseau $y \in \mathbb{R}^r$ est obtenue par une somme pondéré de ces transformations. Ces fonctions splines multivariées sont définies comme suit :

1. Sont des fonctions mathématiques utilisées dans les réseaux de neurones pour introduire la non-linéarité, déterminant si un neurone doit être activé en transformant les valeurs en sortie.
2. B-Spline Neural Networks

Considérant les n_j fonctions de base B-splines $B_{j\,j=1}^{i\,n_i}$ provenant de m espaces splines univariés S_{k_i,t_i}, $i = 1, 2, ..., m$. Alors les fonctions de base de l'espace spline produit tensoriel multivarié $S_{k_1,t_1} \bigotimes S_{k_2,t_2} \bigotimes .... \bigotimes S_{k_m,t_m}$ peuvent être formulées comme suit :

$$S_{i_1,i_2,...,i_m}(x) = \prod_{l=1}^{m} B_{i_l}^{l}(x_l) \quad i_j = 1, 2, ..., n_j, j = 1, 2, ..., m.$$

La sortie du réseau de neurone B-splines est donnée par :

$$y = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} ... \sum_{i_m=1}^{n_m} w_{i_1,i_2,...,i_m} S_{i_1,i_2,...,i_m}$$

où les poids $w \in \mathbb{R}^{r \times n_1 \times ... \times n_1}$ sont les paramètres libres du réseau.
Une représentation graphique de ce réseau est donnée à la figure(3.1)

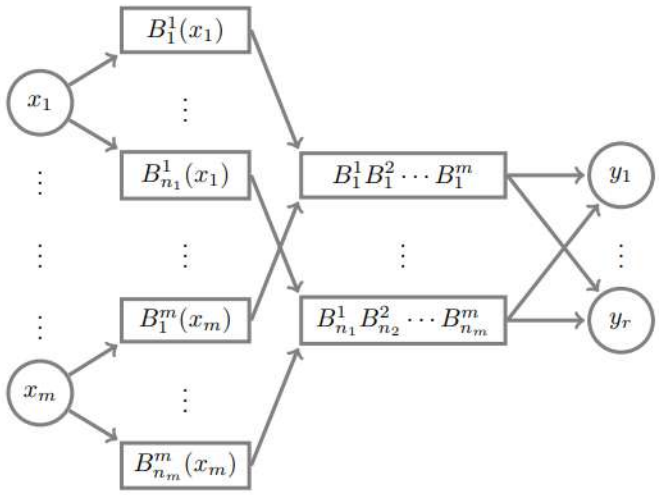

FIGURE 3.1 – Représentation graphique d'un réseau neurone B-splines

3.3 Application sur données réelles

Exemple 3.3.1. Dans cet exemple, nous utilisons un jeu de données (disponibles dans le package "*tidyverse*") pour prédire la valeur médiane des maisons en banlieue de Boston en fonction du pourcentage de la population ayant un statut socio-économique inférieur, tandis la variable *"medv"* représente la valeur médiane des maisons et la variable *"lstat"* signifie le pourcentage de cette population.

Les données sont divisées aléatoirement : 80% pour l'ensemble d'apprentissage et 20% pour l'ensemble de test. Cette division s'assure que le modèle est évalué sur des données qu'il n'a pas vues dans le traitement. Cet exemple s'inscrit dans une démarche typique de machine learning : utilisation de modèles prédictifs(régressions) avec une évaluation de la capacité de généralisation des modèles via cette division.

Nous utilisons trois types de régressions (linéaire, polynomiale et spline) pour comparer les approches de modélisation et déterminer celle qui est la plus adéquate pour prédire "*medv*" à partir de "*lstat*".

- **Préparation des données :**

```r
#  Load the necessary packages
library(caret)
library(MASS)
# Load the data
data("Boston", package = "MASS")
# Split the data into training and test set
set.seed(123)
training.samples <- createDataPartition(Boston$medv, p = 0.8,
    list = FALSE)
train.data  <- Boston[training.samples, ]
test.data <- Boston[-training.samples, ]
```

- **Visualisation des données :**

```r
ggplot(train.data, aes(lstat, medv)) +
  geom_point()
```

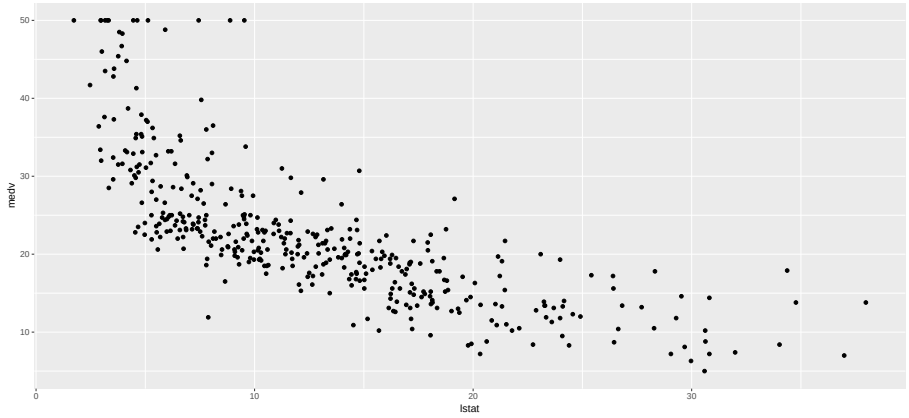

FIGURE 3.2 – Représentation du nuage de points

- **Application de la régression linéaire :**

```r
# Build the model
model <- lm(medv ~ lstat, data = train.data)
# Make predictions
predictions <- predict(model, newdata = test.data)
# Get a summary of the model
summary(model)
```

```
Call:
lm(formula = medv ~ lstat, data = train.data)
Residuals:
    Min      1Q  Median      3Q     Max
-15.218  -4.011  -1.123   2.025  24.459
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  34.6527     0.6230   55.62   <2e-16 ***
lstat        -0.9561     0.0428  -22.34   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 6.144 on 405 degrees of freedom
Multiple R-squared:  0.5521,    Adjusted R-squared:  0.551
F-statistic: 499.2 on 1 and 405 DF,  p-value: < 2.2e-16
```

```r
# Model performance
# Calculate RMSE
rmse <- sqrt(mean((predictions - test.data$medv)^2))

# Calculate  R-squared
sst <- sum((test.data$medv - mean(test.data$medv))^2)
sse <- sum((predictions - test.data$medv)^2)
r2 <- 1 - sse/sst

# Evaluate model performance
performance <- data.frame(
  RMSE = rmse,
  R2 = r2
)
print(performance)
```

```
  RMSE       R2
[1] 6.503817 0.513163
```

Interprétation : Les résultats de la regression linéaire simple indiquent une relation significative entre la variable dépendante *medv* et la variable indépendante *lstat*.

Les coefficients du modèle montre que *lstat* a une influence négative sur la variable *medv* avec un coefficient de -0.9561, ce qui signifie que pour chaque augmentation d'une unité de *lstat* la valeur médiane des maisons diminue en moyenne de 0.9561 unité.

Le coefficient détermination R^2_{adj} de 0.513 et la RMSE de 6.5038 indiquent que le modèle explique environ 51.32% de la variance des valeurs de *medv* avec une erreur moyenne d'environ 6.5 unités.

Donc, le modèle de la régression linéaire est donné sous la forme suivante :

$$medv = 34.6527 - 0.9561 * lstat$$

- **Analyse de variation :**
  Nous utilisons l'ANOVA pour :
  - Vérifier si la variable *lstat* est un prédicteur significatif de la variable *medv*.
  - Comparer la proportion de variance expliquée par le modèle linéaire avec celle des erreurs résiduelles.

```r
anova_results <- anova(model)
print(anova_results)
```

```
 Analysis of Variance Table
Response: medv
           Df Sum Sq Mean Sq F value    Pr(>F)
lstat       1  18840 18839.7  499.16 < 2.2e-16 ***
Residuals 405  15286    37.7
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Interprétation : les résultat de l'ANOVA confirme qu'il existe une relation significative entre les variables *medv* et *lstat*.

- **Visualisation de la régression linéaire :**

```r
ggplot(train.data, aes(lstat, medv)) +
  geom_point() +
  stat_smooth(method = lm, formula = y ~ x)
```

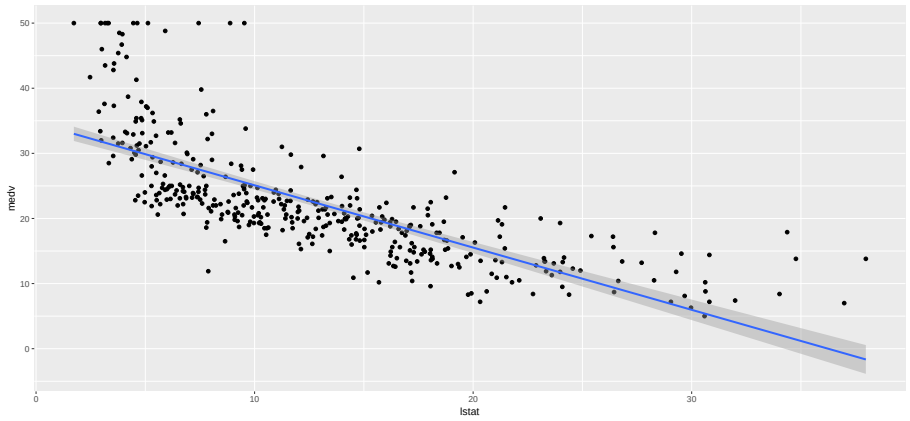

FIGURE 3.3 – Représentation graphique de la régression linéaire

Interprétation : Le graphique de dispersion avec la ligne de la régression montre une relation négative entre les variables.
La dispersion des points autour de la ligne suggère qu'il y a des informations non capturées par ce modèle simple, indiquant qu'il existe une amélioration en utilisant des modèles complexes ou en ajoutant d'autres variables explicatives.

- **Application de la régression polynomiale :**
Comme nous avons citer précédemment qu'il existe d'autres modèles qui modélisent mieux ces données parmi eux la régression polynomiale qui permet de capturer des relations non-linéaires en ajoutant des puissances plus élevées des variables explicatives.
On peut utiliser deux manières différentes pour la déclaration de la régression polynomiale :

```r
lm(medv ~ lstat + I(lstat^2), data = train.data)
```

ou bien

```r
lm(medv ~ poly(lstat, 2, raw = TRUE), data = train.data)
```

```
Call:
lm(formula = medv ~ poly(lstat, 2, raw = TRUE), data = train.data)
Coefficients:
                (Intercept)  poly(lstat, 2, raw = TRUE)1
                    42.5736                      -2.2673
poly(lstat, 2, raw = TRUE)2
                     0.0412
```

Interprétation des résultats : Ces résultats montrent qu'il existe une relation complexe entre la variable réponse *medv* et la variable explicative *lstat*.

Le coefficient de -2.2673 indique qu'une augmentation d'une unité de la variable *lstat* entraîne une diminution de 2.2673 unités de *medv* tandis que, le coefficient de terme quadratique (0.0421) accentue la relation non linéaire entre ces deux variables. Ainsi ce modèle, dicte que la variable expliquée *medv* est influencée a la fois linéairement et quadratiquement par la variable *lstat*.

Donc, le modèle est défini comme suit :

$$medv = 42.5736 - 2.2673 * lstat + 0.0412 * lstat^2$$

- **Visualisation du modèle de régression :**

```
ggplot(train.data, aes(lstat, medv)) +
  geom_point() +
  stat_smooth(method = lm, formula = y ~ poly(x, 2, raw = TRUE)
    )
```

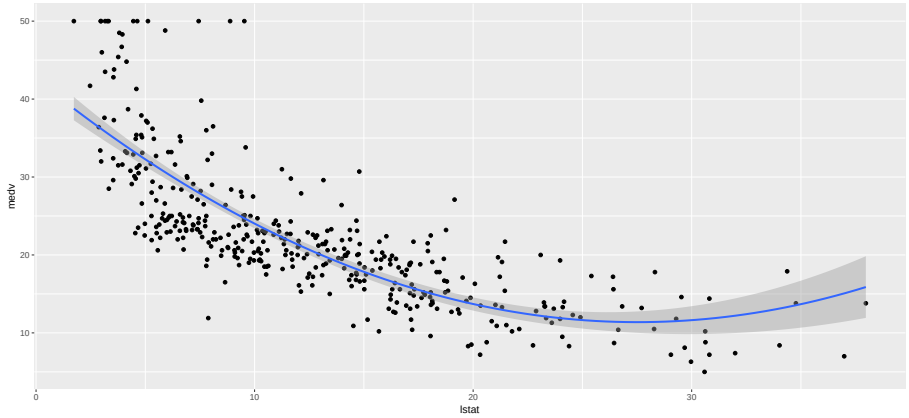

FIGURE 3.4 – Représentation graphique de la régression polynomiale.

Interprétation : Ce graphique confirme l'existence d'une relation complexe et non linéaire entre les variables *medv* et *lstat*. On observe qu'au début, une augmentation de la variable explicative provoque une diminution de la variable dépendante. Cependant, après un certain seuil, cette diminution se stabilise et peut même montrer une légère inversion pour des valeurs plus élevées de *lstat*. Cette courbe non linéaire capture mieux les variations des données par rapport à une simple régression linéaire.

- **Application de la régression spline :**
  Nous allons créer un modèle à l'aide d'une spline cubique (i.e n=3)

```r
library(splines)
#Build the model
knots <- quantile(train.data$lstat, p = c(0.25, 0.5, 0.75))
model <- lm(medv ~ bs(lstat, knots = knots), data = train.data)

# Make predictions
predictions <- predict(model, newdata = test.data)

# Calculate RMSE
rmse <- sqrt(mean((predictions - test.data$medv)^2))

# Calculate  R-squared
sst <- sum((test.data$medv - mean(test.data$medv))^2)
sse <- sum((predictions - test.data$medv)^2)
r2 <- 1 - sse/sst

# Evaluate model performance
performance <- data.frame(
  RMSE = rmse,
  R2 = r2
)

#  Display performance
print(performance)
```

```
      RMSE        R2
[1] 5.317372 0.6741241
```

Interprétation des résultats : Ces deux indicateurs indiquent que le modèle de régression spline a une performance raisonnable avec une erreur moyenne de prédiction d'environ 5.32, ce qui indique que la dispersion des valeurs prédictives autour des valeurs observées. Tandis que le coefficient de détermination R^2 de valeur 0.674121 signifie que 67.41% de la variance des données est expliquée par le modèle, ce qui est généralement considéré comme une performance respectable.

- **Visualisation du modèle de régression :**

```r
ggplot(train.data, aes(lstat, medv) ) +
  geom_point() +
  stat_smooth(method = lm, formula = y ~ splines::bs(x, df = 3)
     )
```

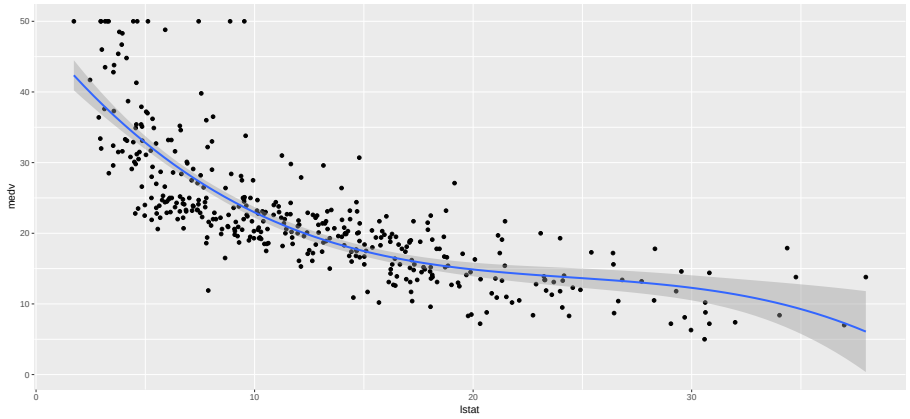

FIGURE 3.5 – Représentation graphique de la régression spline.

Interprétation du graphique : En remarquant que la courbe montre une tendance décroissante générale, indiquant une relation négative entre les variables, on remarque aussi que la courbe suit les points de données de manière flexible, ce qui signifie que le modèle peut s'adapter aux variations subtiles.

La bande en gris autour de la courbe représente l'intervalle de confiance des prédictions qui montre la variabilité des prédictions du modèle (une bande plus étroite indique une plus grande précision de prédictions tandis qu'une bande plus large indique plus d'incertitude.)

Conclusion : À partir des résultats des performances de ces différents modèles ainsi leurs graphiques, on peut dire que la régression spline est le modèle le plus efficace pour capturer les relations complexes et non-linéaires.

Conclusion générale

Comme nous l'avons mentionné en introduction, ce travail a mis en évidence l'importance de l'outil statistique dans le *machine learning*. En effet, la partie essentielle du *machine learning* est d'abord de nature mathématique, en particulier statistique, avant que des méthodes informatiques puissantes conçoivent les programmes et les logiciels d'application. Le *machine learning* est une étape importante qui rend la prédiction efficace en intelligence artificielle.

A travers ce travail, nous avons montré l'utilité des *méthodes de régression spline* qui ont pour principe de pallier à la complexité des grandes masses de données qu'il faut traiter quand la prédictivité est nécessaire.
En effet, la statistique est un outil d'aide à la décision puissant qui exige de la précision dans la maîtrise des données observées. Tout cela passe par un ajustement des données le plus adéquat possible. C'est en ce sens que la régression spline joue un rôle central dans la modélisation. Associer un modèle statistique à des données complexes est une tâche difficile que la régression Spline arrive à simplifier. De plus, l'usage du puissant langage R permet de mettre en œuvre les techniques ainsi construites.

Pour améliorer ce travail, il serait souhaitable d'explorer d'autres branches de la statistique qui peuvent contribuer à une analyse plus approfondie des données et rendre ainsi plus efficace la maîtrise des outils d'aide à la décision en *machine learning*.

Bibliographie

[1] Alimrina.K, (2022), *Machine Learning et Application en finance*. Mémoire de fin de cycle. Université A. Mira de Béjaïa, Faculté des Sciences Exactes, Département de Recherche Opérationnelle.

[2] Amroun.S, (2011), *L'estimation de la Courbe De Régression De La Moyenne*. Mémoire de Magister, Université A. Mira de Béjaïa, Faculté des Sciences Exactes, Département de Recherche Opérationnelle.

[3] Bekda.L, (2017), *Approche Bayésienne dans les modèles de régression.*. Mémoire de Master(PS). Université Mouloud Mammeri, Tizi-Ouzou, Faculté des Sciences , Département de Mathématiques.

[4] Besse.P, (2006), *Apprentissage statistique & Data mining*. Cours de Master. Institut de Mathematiques de Toulouse, Laboratoire de Statistique et Probabilites — UMR CNRS C5583, Institut National des Sciences Appliquees de Toulouse — 31077 – Toulouse cedex 4.

[5] Campolucci.P, Capperelli.F, Guarnieri.S, Piazza.F et Uncini.A, (1996), *Neural Networks with Spline Activation Function*. Proceedings of 8th Mediterranean Electrotechnical Conference on Industrial Applications in Power Systems, Computer Science and Telecommunications (MELECON 96)

[6] Chavent.M, (2013), *Chapitre II Régression linéaire multiple*. Licence 3 MIASHS-Université de Bordeaux.

[7] Curry.H.B et Schoenberg. I.J, (1966),*On Polya Frequency Functions, IV : The Fundamental Spline Functions and their Limits*. Journal d'analyse mathématique, 17 (1), 71-107.

[8] DRIS.L et HACHEMI.W, (2016), *Régression linéaire multiple et modèle linéaire général*. Université A. Mira de Béjaïa, Faculté des Sciences Exactes, Département de Mathématiques.

[9] Fellag.H, (2023), *Modèles de régression*. Cours Master2(PS). Université Mouloud Mammeri, Tizi-Ouzou, Faculté des Sciences , Département de Mathématiques.

[10] Fellag.H, (2023), *Stratégie numérique et Intelligence artificielle dans le management*. Cours Master2(PS). Université Mouloud Mammeri, Tizi-Ouzou, Faculté des Sciences , Département de Mathématiques.

[11] Galton. F, (1886). *Regression Towards Mediocrity in Hereditary Stature*. Journal of the Anthropological Institute of Great Britain and Ireland, 15, 246-263.

[12] Kassambara.A,(2018),*Machine Learning Essentials Pratical Guide in R*. Edition 1. sthda.com/english.

[13] Masse.J.C, *Interpolation et lissage*. Document de travail. Université Laval, Département de mathématiques et des statistiques.

[14] Pansu,P.(2004), *Courbes B-splines*, Document de travail. Université Paris-Saclay.

[15] Patenaude,V.(2011), *Utilisation de splines monotones afin de condenser des tables de mortalité dans un contexte bayésien.* Mémoire présenté à la Faculté des études supérieures en vue de l'obtention du grade de Maître ès sciences (M.Sc.) en Statistique. Université de Montréal, Faculté des arts et des sciences, Département de mathématiques et de statistique.

[16] Perperoglou,A, Sauerbrei.W, Abrahamowicz.M et Schmid.M. (2019), *A Review of Spline Function Procedures in R.* BMC Medical Research Methodology

[17] Reinsch,C.H, (1967). *Smoothing by Spline Functions.* Numerische Mathematik, 10 (3), 177-183.

[18] Samuel,A,L, (1959). *Some studies in machine learning using the game of chechers.* IBM journal of research and development, 44(1.2) :206–226.

[19] Scardapane.S, Scarpiniti.M , Comminiello.D et Uncini.A. (2016), *Learning activation functions from data using cubic spline interpolation*, ArXiv e-prints arXiv :1605.05509.

[20] Schoenberg,I.J,(1964),*Spline functions and the problem of graduation.* Mathematics, 52,974-50

[21] Thomas-Agnan,C. *Estimateurs splines.* Document de travail, Toulouse School of Economics (TSE). 2006.

[22] Toulouse,J. (2023), *Outils et méthodes mathématiques (LU2CI007).* Cours dispensé au Laboratoire de Chimie Théorique, Sorbonne Université and CNRS, 75005 Paris, France. Institut Universitaire de France, F-75005 Paris, France.

[23] Wang,K.(2013), *A study of cubic spline interpolation.* Insight : Rivier Academic Journal, volume9(2).

[24] Whittaker,E.T,(1923). *On a new method of graduation.* Proceedings of the Edinburgh Mathematical Society, vol. 41, pp. 6375.

[25] ZERDANI,O.(2015/2016),*Cours sur les Méthodes Numériques*, Cours L2. Université Mouloud Mammeri, Tizi-Ouzou, Faculté des Sciences, Département de Mathématiques

Résumé

Dans ce travail, nous avons étudier les différents modèles de régression (paramétrique et non-paramétrique), ces principaux concepts avec des exemples d'application, puis nous avons étudier la fonction spline afin de mener à la régression spline. Enfin, nous avons donner un exemple d'application sur le modèle de régression spline dans le cadre du machine learning.

Mots clés : Régression spline, fonction spline, interpolation polynomiale, apprentissage automatique, lissage.

Abstract

In this work, we studied different regression models, both parametric and non-parametric, exploring their main concepts through application examples. We then examined spline functions, leading to spline regression.
Finally, we provided an example of the application of the spline regression model within the framework of machine learning.

Keywords : Spline regression, spline function, Polynomial interpolation, machine learning, smoothing.