MTH 8302 - Modèles de Régression et d'Analyse de Variance

Leçon 2 (Clarifications) : Estimation par Maximum de Vraisemblance et intervalles de Confiance et de Prédictions en Régression Linéaire Multiple

Polytechnique Montréal - Hiver 2025

Chiheb Trabelsi

March 10, 2025

Table des Matières

Estimation par Maximum de Vraisemblance en Régression Linéaire Multiple

Estimation par Maximum de Vraisemblance en Régression Linéaire Simple

Estimation par Maximum de Vraisemblance en Rég Simple

- Nous allons démontrer l'expression de l'estimateur du maximum de vraisemblance (MV) en régression linéaire multiple, en partant du cas de la régression linéaire simple et en établissant le lien entre les deux.
- Dans la régression linéaire simple, on considère le modèle :

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i, \quad \text{où} \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2).$$

- Chaque Y_i suit une loi normale de moyenne $\mathbb{E}[Y_i] = \beta_0 + \beta_1 X_i$ et de variance σ^2 :

$$Y_i \sim \mathcal{N}(\beta_0 + \beta_1 X_i, \sigma^2).$$

- Puisque les erreurs sont indépendantes, la vraisemblance pour un échantillon de taille n est donnée par :

$$L(\beta_0, \beta_1, \sigma^2) = p(Y_1, Y_2, \ldots, Y_n \mid \boldsymbol{\beta}, \sigma^2) = \prod_{i=1}^{n} p(Y_i \mid \boldsymbol{\beta}, \sigma^2).$$

$$\Rightarrow L(\beta_0, \beta_1, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\overbrace{(Y_i - \beta_0 - \beta_1 X_i)}^{\epsilon_i}{}^2}{2\sigma^2} \right).$$

Estimation par Maximum de Vraisemblance en Rég Simple

- En prenant le logarithme, on obtient la log-vraisemblance :

$$\log L(\beta_0, \beta_1, \sigma^2) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - \beta_0 - \beta_1 X_i)^2.$$

- Maximiser $\log L$ revient à minimiser la somme des carrés des erreurs:

$$\sum_{i=1}^{n}(Y_i - \beta_0 - \beta_1 X_i)^2.$$

- On reconnaît ici le critère des moindres carrés ordinaires (MCO), qui donne les estimateurs :

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^{n}(X_i - \bar{X})^2}, \quad \hat{\beta}_0 = \bar{Y} - \hat{\beta}_1\bar{X}.$$

- Ainsi, en régression linéaire simple, les estimateurs du maximum de vraisemblance sont équivalents aux moindres carrés.

Estimation par Maximum de Vraisemblance en Régression Linéaire Multiple

Estimation par Maximum de Vraisemblance en Rég Mul

- Dans le cas général de la régression linéaire multiple, on considère le modèle :

$$\mathbf{Y} = \underbrace{\mathbf{X}\boldsymbol{\beta}}_{\hat{\mathbf{Y}}} + \boldsymbol{\epsilon}, \quad \text{où} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 I_n).$$

- Ici :
  - $\mathbf{Y}$ est le vecteur des observations $(n \times 1)$.
  - $\hat{\mathbf{Y}}$ est le vecteur des prédictions $(n \times 1)$.
  - $\mathbf{X}$ est la matrice des prédicteurs $(n \times p)$.
  - $\boldsymbol{\beta}$ est le vecteur des paramètres $(p \times 1)$.
  - $\boldsymbol{\epsilon}$ est un vecteur de bruit blanc gaussien.
- Chaque Y_i suit une distribution normale :

$$Y_i \sim \mathcal{N}(X_i\boldsymbol{\beta}, \sigma^2) \Rightarrow \mathbf{Y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \boldsymbol{\Sigma} = \sigma^2\mathbf{I}_n).$$

Estimation par Maximum de Vraisemblance en Rég Mul

- **Vecteur des observations :**

$$\mathbf{Y} = \begin{bmatrix} Y_1, & Y_2, & \cdots, & Y_n \end{bmatrix}^\top \in \mathbb{R}^n.$$

- **Matrice des prédicteurs :** Chaque ligne $\mathbf{X}_i$ représente les valeurs des prédicteurs pour l'observation i :

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1, & \mathbf{X}_2, & \cdots, & \mathbf{X}_n \end{bmatrix}^\top \in \mathbb{R}^{n\times(p+1)}.$$

- **Vecteur des coefficients :**

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0, & \beta_1, & \cdots, & \beta_p, & \end{bmatrix} \in \mathbb{R}^{(p+1)}.$$

- **Prédiction du modèle :** Le vecteur aléatoire des prédictions $\hat{\mathbf{Y}}$ valeur prédite est donnée par le produit matriciel $\mathbf{X}\boldsymbol{\beta}$:

$$\hat{\mathbf{Y}} = \begin{bmatrix} \hat{Y}_1, & \hat{Y}_2, & \cdots, & \hat{Y}_n \end{bmatrix}^\top = \begin{bmatrix} \mathbf{X}_1\boldsymbol{\beta}, & \mathbf{X}_2\boldsymbol{\beta}, & \cdots, & \mathbf{X}_n\boldsymbol{\beta} \end{bmatrix}^\top = \overbrace{\mathbf{X}\boldsymbol{\beta}}^{\in\mathbb{R}^n}.$$

- **Vecteur des erreurs du modèle de régression :**

$$\boldsymbol{\epsilon} = \mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{Y} - \mathbf{X}\boldsymbol{\beta} = \begin{bmatrix} Y_1 - \mathbf{X}_1\boldsymbol{\beta}, & Y_2 - \mathbf{X}_2\boldsymbol{\beta}, & \cdots, & Y_n - \mathbf{X}_n\boldsymbol{\beta} \end{bmatrix}^\top.$$

Estimation par Maximum de Vraisemblance en Rég Mul

- L'hypothèse d'indépendance des erreurs nous permet d'écrire la vraisemblance sous la forme d'un produit :
$L(\boldsymbol{\beta}, \sigma^2) = p(\mathbf{Y} \mid \boldsymbol{\beta}, \sigma^2)$

$$= p(Y_1, Y_2, \ldots, Y_n \mid \boldsymbol{\beta}, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\overbrace{(Y_i - \mathbf{X}_i\boldsymbol{\beta})}^{\epsilon_i}{}^2}{2\sigma^2} \right).$$

- En exploitant $\prod_{i=1}^{n} \exp(a_i) = \exp\left(\sum_{i=1}^{n} a_i\right)$, on obtient :

$$L(\boldsymbol{\beta}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} \overbrace{(Y_i - \mathbf{X}_i\boldsymbol{\beta})}^{\epsilon_i}{}^2 \right).$$

- La somme des carrés des erreurs est le produit scalaire :

$$\sum_{i=1}^{n} (\epsilon_i)^2 = \sum_{i=1}^{n} (Y_i - \mathbf{X}_i\boldsymbol{\beta})^2 = \sum_{i=1}^{n} \boldsymbol{\epsilon}^\top \boldsymbol{\epsilon} = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}).$$

- Ainsi, en réécrivant sous forme matricielle, on obtient :

$$L(\boldsymbol{\beta}, \sigma^2) = (2\pi\sigma^2)^{-n/2} \exp\left(-(1/2\sigma^2)(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \right).$$

Estimation par Maximum de Vraisemblance en Rég Mul

- Maximiser la log-vraisemblance est équivalent à maximiser la vraisemblance puisque la function $\log$ est une function monotone. En prenant le logarithme, on obtient la log-vraisemblance :

$$\begin{aligned}\log\left[p(\mathbf{Y} \mid \boldsymbol{\beta}, \sigma^2)\right] &= \log\left[L(\boldsymbol{\beta}, \sigma^2)\right] = l(\boldsymbol{\beta}, \sigma^2) \\ &= -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}).\end{aligned}$$

- Maximiser la log-vraisemblance revient à trouver l'argument de $\boldsymbol{\beta}$ qui la maximise :

$$\begin{aligned}\arg_{\boldsymbol{\beta}}\max l(\boldsymbol{\beta}, \sigma^2) &= \arg_{\boldsymbol{\beta}}\max -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \\ &= \arg_{\boldsymbol{\beta}}\max -(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \\ &= \arg_{\boldsymbol{\beta}}\min +(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})\end{aligned}$$

- Comme dans le cas de la régression linéaire simple, maximiser la log-vraisemblance revient à trouver l'argument de $\boldsymbol{\beta}$ qui minimise la somme des carrés des erreurs :

$$\hat{\boldsymbol{\beta}}_{\mathsf{MV}} = \arg_{\boldsymbol{\beta}}\min(\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}) = \arg_{\boldsymbol{\beta}}\min(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) = \hat{\boldsymbol{\beta}}_{\mathsf{MCO}}.$$

Estimation par Maximum de Vraisemblance en Rég Mul

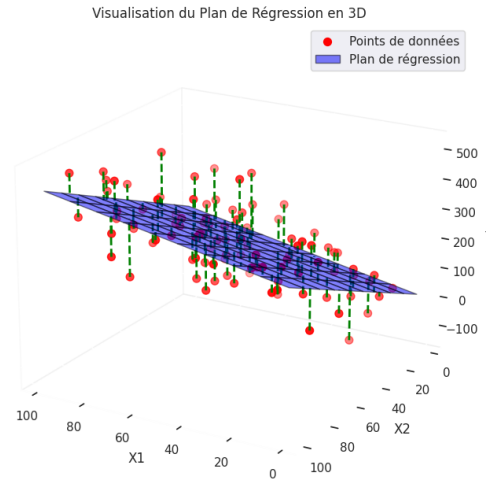
- On retrouve ici le critère des moindres carrés ordinaires (MCO), qui donne les estimateurs :

$$\hat{\boldsymbol{\beta}}_{\text{MV}} = \arg_{\boldsymbol{\beta}} \min \left(\log \left[p(\mathbf{Y} = \boldsymbol{y} \mid \boldsymbol{\beta}, \sigma^2) \right] \right) = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \boldsymbol{y} = \hat{\boldsymbol{\beta}}_{\text{MCO}},$$

  où $\boldsymbol{y}$ est la valeur observée du vecteur aléatoire $\mathbf{Y}$
- Maximiser donc la vraisemblance revient à minimiser la somme des carrés des erreurs, ce qui explique pourquoi les estimateurs MV et MCO coïncident sous l'hypothèse de normalité et d'indépendance des erreurs.

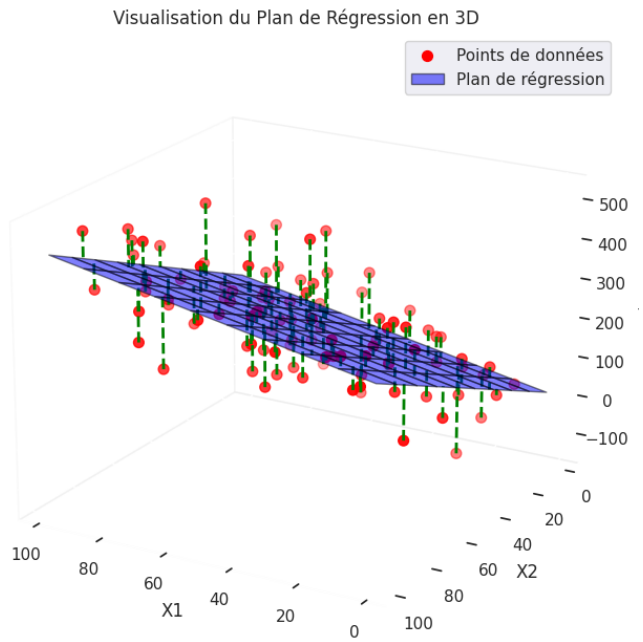
Estimation par Maximum de Vraisemblance en Rég Mul

- **Interprétation Géométrique :**
  - **Projection orthogonale sur le sous-espace engendré par les prédicteurs:**
  - L'hyperplan en bleu représente la **régression linéaire multiple**, c'est-à-dire le sous-espace sur lequel les valeurs prédites $\hat{y}$ sont projetées ($\hat{\boldsymbol{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$).
  - Les **lignes verticales vertes pointillées** représentent les **résidus** $\boldsymbol{e} = \boldsymbol{y} - \hat{\boldsymbol{y}}$, c'est-à-dire la différence entre les points de données réels (rouges) et leurs projections sur l'hyperplan (prédictions).
  - La **minimisation des moindres carrés** ajuste cet hyperplan de manière à minimiser la somme des carrés des longueurs de ces segments.



Estimation par Maximum de Vraisemblance en Rég Mul

- Sous l'hypothèse que les erreurs sont indépendantes et suivent une loi normale $\mathcal{N}(0, \sigma^2)$, la **log-vraisemblance** est maximisée lorsque la somme des carrés des distances entre les **points rouges** (observations) et le **plan bleu** (prédictions) est minimisée.
- Comme dans l'interprétation des MCO, l'objectif du MV est également de **minimiser ces écarts quadratiques**.



Pourquoi ne pas chercher la dérivée seconde ?

- Dans les deux méthodes (MCO et MV), la function de coût à minimiser une fonction quadratique en $\boldsymbol{\beta}$.

$$C(\boldsymbol{\beta}) = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}).$$

- Pour s'assurer qu'un minimum est bien atteint, on pourrait vérifier la matrice Hessienne

$$\nabla C(\boldsymbol{\beta}) = \frac{\partial C}{\partial \boldsymbol{\beta}} = -2\mathbf{X}^{\mathrm{T}}\mathbf{Y} + 2\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta} \Rightarrow \nabla^2 C(\boldsymbol{\beta}) = 2\mathbf{X}^\top\mathbf{X}.$$

- La matrice de Gram $\mathbf{X}^\top\mathbf{X}$ est semi-définie positive. Si les colonnes de $\mathbf{X}$ sont linéairement indépendantes, alors $\mathbf{X}^\top\mathbf{X}$ est définie positive. Cela garantit que la fonction est convexe et atteint un minimum unique.
- **Cette indépendance linéaire, et par conséquent la convexité de $\mathbf{X}^\top\mathbf{X}$, est garantie par l'hypothèse d'absence de multicolinéarité des colonnes de $\mathbf{X}$.**
- **Conclusion :** La convexité de la solution est garantie. La Hessienne est constante et elle est définie positive dans l'absence de la multicolinéarité des colonnes de $\mathbf{X}$.

Table des Matières

Intervalle de Confiance pour la Réponse Moyenne et Intervalle de Prédiction pour une nouvelle Observation en Régression Linéaire Multiple

Clarifier la Logique Sous-Jacente

Comment peut-on quantifier notre incertitude sur une estimation?

Réponse intuitive :

- On exprime ce que l'on cherche sous forme de probabilité.
- On utilise l'erreur standard pour formaliser cette incertitude.
- On en déduit un intervalle, basé sur la distribution de l'estimateur.

Intervalle de Confiance pour la Réponse Moyenne en Régression Linéaire Multiple

Intervalle de Confiance : Probabilité et Erreur Standard

- Lorsqu'on veut quantifier l'incertitude sur la réponse moyenne du modèle pour une valeur observée des données $\mathbb{E}[Y \mid \mathbf{X_0} = \boldsymbol{x_0}]$, nous essayons d'encadrer $\mathbb{E}[Y \mid \mathbf{X_0} = \boldsymbol{x_0}]$ avec une formulation probabiliste.
- On note Y_0 la réponse du modèle pour une valeur observée des données $\mathbf{X_0} = \boldsymbol{x_0}$. Nous pouvons noter $\mathbb{E}[Y \mid \mathbf{X_0} = \boldsymbol{x_0}] = \mathbb{E}[Y_0]$.
- **Formulation Probabiliste :** Nous avons vu dans le cours que

$$\frac{\hat{Y}_0 - \mathbb{E}[Y_0]}{\text{SE}(\hat{Y}_0)} = \frac{\hat{Y}_0 - \mathbb{E}[Y \mid \mathbf{X_0} = \boldsymbol{x_0}]}{s\sqrt{\mathbf{x}_0^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_0}} \sim t_{n-p-1}.$$

  - Où $\text{SE}(\hat{Y}_0) = s\sqrt{\mathbf{x}_0^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_0}$ est **l'erreur type de** $\hat{Y}_0$
- Cela nous permet d'encadrer $\mathbb{E}[Y \mid \mathbf{X_0} = \boldsymbol{x_0}] = \mathbb{E}[Y_0]$ entre 2 valeurs critiques de la distribution t déterminées par un certain niveau de signification α et le degré de liberté $n - p - 1$:

$$P\left(-t_{\alpha/2,n-p-1} \leq \frac{\hat{Y}_0 - \mathbb{E}[Y_0]}{\text{SE}(\hat{Y}_0)} \leq t_{\alpha/2,n-p-1}\right) = 1 - \alpha.$$

Intervalle de Confiance : Probabilité et Erreur Standard

- Étant donné :

$$P\left(-t_{\alpha/2,n-p-1} \leq \frac{\hat{Y}_0 - \mathbb{E}[Y_0]}{\mathsf{SE}(\hat{Y}_0)} \leq t_{\alpha/2,n-p-1}\right) = 1-\alpha,$$

  nous encadrons alors $\mathbb{E}[Y_0]$ en utilisant une formulation probabiliste : $P\left(\hat{Y}_0 - \mathsf{SE}(\hat{Y}_0) \times t_{\alpha/2,n-p-1} \leq \mathbb{E}[Y_0] \leq \hat{Y}_0 + \mathsf{SE}(\hat{Y}_0) \times t_{\alpha/2,n-p-1}\right) = 1-\alpha$.

- $\Rightarrow$ Avec un niveau de confiance $1-\alpha$, on a :

$$\mathbb{E}[Y_0] \in \left[\hat{Y}_0 - \mathsf{SE}(\hat{Y}_0) \times t_{\alpha/2,n-p-1}, \hat{Y}_0 + \mathsf{SE}(\hat{Y}_0) \times t_{\alpha/2,n-p-1}\right]$$

- L'intervalle de confiance à $(1-\alpha)\%$ est alors donné par:

$$\hat{Y}_0 \pm t_{\alpha/2,n-p-1} \times \mathsf{SE}(\hat{Y}_0).$$

Intervalle de Prédiction pour une Nouvelle Observation en Régression Linéaire Multiple

Intervalle de Prédiction : Probabilité et Erreur Standard

- Lorsqu'on veut quantifier l'incertitude pour, non pas la moyenne de la réponse du modèle mais pour la réponse Y_0 tout court étant donné qu'on observe une valeur particulière d'une nouvelle observation $\mathbf{X_0} = \boldsymbol{x_0}$, nous essaierons alors d'encadrer Y_0 avec une formulation probabiliste.
- On note Y_0 la nouvelle réponse du modèle pour une valeur observée des données $\mathbf{X_0} = \boldsymbol{x_0}$.
- **Formulation Probabiliste :** Nous avons vu dans le cours que

$$\frac{Y_0 - \hat{Y}_0}{\text{SE}(Y_0 - \hat{Y}_0)} = \frac{Y_0 - \hat{Y}_0}{s\sqrt{1 + \mathbf{x}_0^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_0}} \sim t_{n-p-1}.$$

  - Où $\text{SE}(Y_0 - \hat{Y}_0) = s\sqrt{\mathbf{x}_0^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_0}$ est **l'erreur type de** $Y_0 - \hat{Y}_0$.
- Cela nous permet d'encadrer Y_0 entre 2 valeurs critiques de la distribution t déterminées par un certain niveau de signification α et le degré de liberté $n - p - 1$:

$$P\left(-t_{\alpha/2,n-p-1} \leq \frac{Y_0 - \hat{Y}_0}{\text{SE}(Y_0 - \hat{Y}_0)} \leq t_{\alpha/2,n-p-1}\right) = 1 - \alpha.$$

Intervalle de Prédiction : Probabilité et Erreur Standard

- Étant donné :

$$P\left(-t_{\alpha/2,n-p-1} \leq \frac{Y_0 - \hat{Y}_0}{\mathsf{SE}(Y_0)} \leq t_{\alpha/2,n-p-1}\right) = 1-\alpha,$$

nous encadrons alors Y_0 en utilisant une formulation probabiliste :
$P\left(\hat{Y}_0 - \mathsf{SE}(Y_0) \times t_{\alpha/2,n-p-1} \leq Y_0 \leq \hat{Y}_0 + \mathsf{SE}(Y_0) \times t_{\alpha/2,n-p-1}\right) = 1-\alpha.$

- $\Rightarrow$ Avec un niveau de confiance $1-\alpha$, on a :

$$Y_0 \in \left[\hat{Y}_0 - \mathsf{SE}(Y_0) \times t_{\alpha/2,n-p-1}, \hat{Y}_0 + \mathsf{SE}(Y_0) \times t_{\alpha/2,n-p-1}\right]$$

- L'intervalle de prédiction à $(1-\alpha)\%$ est alors donné par:

$$\hat{Y}_0 \pm t_{\alpha/2,n-p-1} \times \mathsf{SE}(Y_0).$$

Comparaison : Intervalle de Confiance vs Intervalle de Prédiction

- **Intervalle de Confiance (IC) :**
  - Encadre **la moyenne** $\mathbb{E}[Y|X_0 = x_0]$.
  - Reflète l'incertitude sur la moyenne prédisée.
  - **Devient plus étroit** si n augmente.
- **Intervalle de Prédiction (IP) :**
  - Encadre une **nouvelle observation** Y_0.
  - Intègre la variabilité des nouvelles observations.
  - **Toujours plus large que IC** car il inclut l'incertitude sur Y_0.
  - L'intervalle de confiance ne tient compte que de l'incertitude sur la moyenne estimée (le fait que nous n'avons qu'un échantillon fini pour faire une estimation de la vraie moyenne).
  - L'intervalle de prédiction, en revanche, inclut la variabilité supplémentaire des observations individuelles, ce qui le rend plus large.

Interprétation des IC et IP

- Dans les graphiques générés, nous avons tracé deux types d'intervalles autour de la courbe de régression linéaire :
  - **L'intervalle de confiance (IC - en vert)** : Il représente la plage dans laquelle nous estimons que la moyenne conditionnelle de la variable cible (valeur médiane des logements, MEDV) se trouve pour une valeur donnée de RM (nombre moyen de pièces). Plus précisément, il nous donne une estimation de la moyenne de tous les logements ayant un nombre donné de pièces, avec un niveau de confiance de 95%.
  - **L'intervalle de prédiction (IP - en rouge)** : Il représente la plage dans laquelle une nouvelle observation individuelle est susceptible de se trouver pour une valeur donnée de RM. Comme il prend en compte non seulement l'incertitude du modèle mais aussi la variabilité intrinsèque des logements, cet intervalle est toujours plus large que l'intervalle de confiance.

Rég Simple : Intervalles de Confiance et de Prédiction

```python
# Importation des bibliothèques nécessaires
import numpy as np
import pandas as pd
import scipy.stats as stats
import matplotlib.pyplot as plt
import statsmodels.api as sm
import seaborn as sns
from ISLP import load_data
sns.set_theme()
# Chargement des données Boston Housing
Boston = load_data('Boston')
# Sélection de la variable explicative (Nombre de chambres : RM) et la variable cible (MEDV)
X = Boston[['rm']]
Y = Boston['medv'].values  # Prix médian des logements
# Ajout d'une colonne d'intercept
X_with_intercept = sm.add_constant(X)
# Ajustement du modèle de régression linéaire simple
model = sm.OLS(Y, X_with_intercept).fit()
# Prédictions sur l'échantillon
Y_pred = model.predict(X_with_intercept)
```

Rég Simple : Intervalles de Confiance et de Prédiction

```
# Calcul des paramètres statistiques
n = len(X)
s_squared = np.sum((Y - Y_pred) ** 2) / (n - 2)
s = np.sqrt(s_squared)
t_value = stats.t.ppf(1 - 0.025, df=n - 2)
# Matrice (X'X)^-1 pour RM
XTX_inv = np.linalg.inv(X_with_intercept.T @ X_with_intercept)
# Définition d'une plage de valeurs pour RM
rm_values = np.linspace(X['rm'].min(), X['rm'].max(), 100)
X_range = sm.add_constant(pd.DataFrame({'rm': rm_values}))
# Prédictions et calcul des marges IC et IP
IC_inf, IC_sup, IP_inf, IP_sup = [], [], [], []
for x_0 in X_range.values:
    variance_IC = s_squared * (x_0 @ XTX_inv @ x_0.T)
    variance_IP = s_squared * (1 + x_0 @ XTX_inv @ x_0.T)
    SE_IC = np.sqrt(variance_IC)
    SE_IP = np.sqrt(variance_IP)
    y_hat = model.predict([x_0])[0]
    IC_inf.append(y_hat - t_value * SE_IC)
    IC_sup.append(y_hat + t_value * SE_IC)
    IP_inf.append(y_hat - t_value * SE_IP)
    IP_sup.append(y_hat + t_value * SE_IP)
# Définition de points de test pour RM = 5, 6, 7, 8
rm_test_values = np.array([5, 6, 7, 8])
test_points = sm.add_constant(pd.DataFrame({'rm': rm_test_values}))
test_pred = model.predict(test_points) # Prédictions pour ces points
test_IC = []
test_IP = []
```

Rég Simple : Intervalles de Confiance et de Prédiction

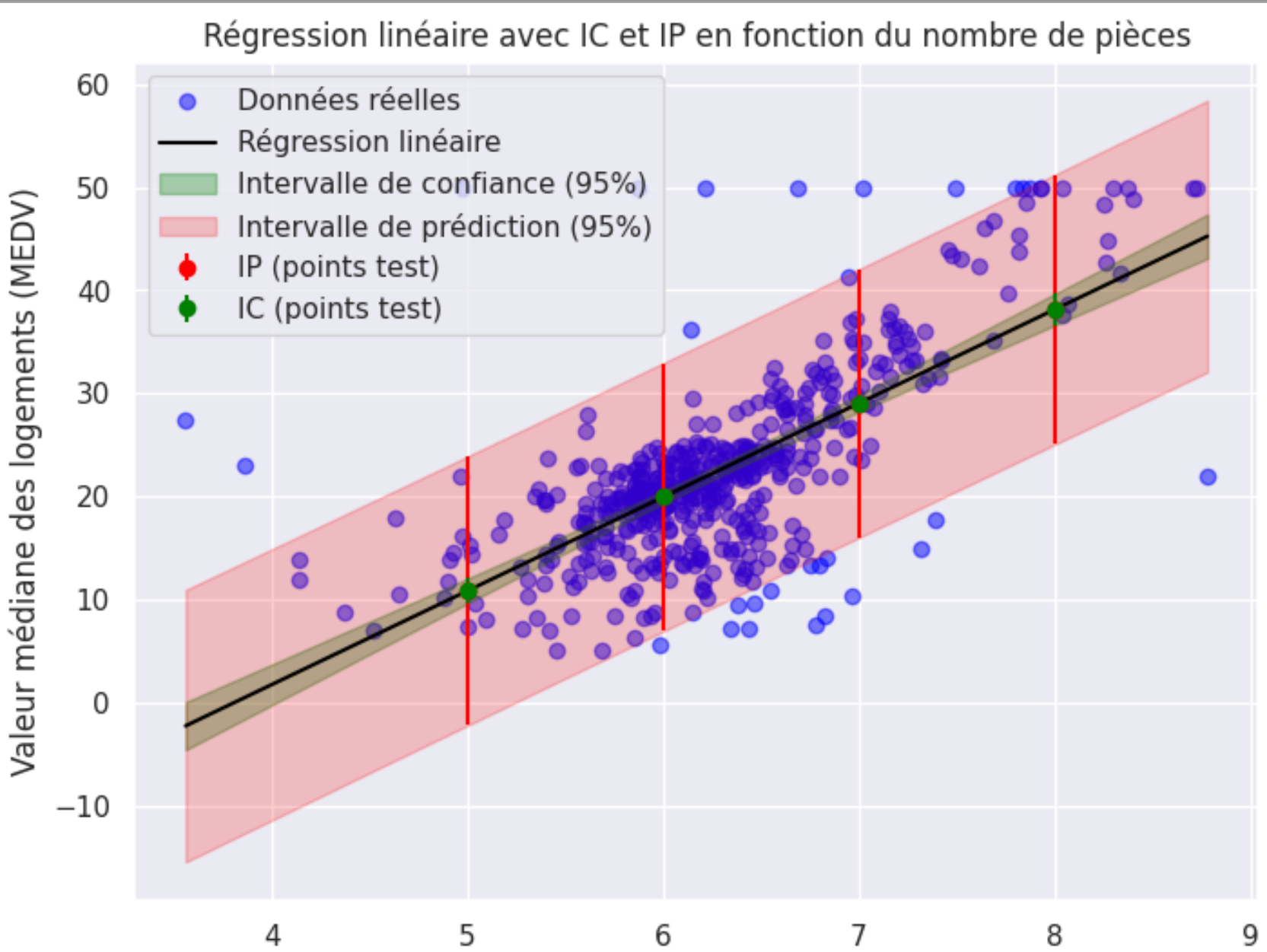
```
for x_0 in test_points.values:
    variance_IC = s_squared * (x_0 @ XTX_inv @ x_0.T)
    variance_IP = s_squared * (1 + x_0 @ XTX_inv @ x_0.T)
    SE_IC = np.sqrt(variance_IC)
    SE_IP = np.sqrt(variance_IP)
    test_IC.append(t_value * SE_IC)
    test_IP.append(t_value * SE_IP)
for i, rm_val in enumerate(rm_test_values): # Affichage des résultats
    print(f"Prédiction pour RM = {rm_val} : {test_pred[i]:.2f}")
    print(f"IC à 95%: [{test_pred[i] - test_IC[i]:.2f}, {test_pred[i] + test_IC[i]:.2f}]")
    print(f"Ip à 95%: [{test_pred[i] - test_IP[i]:.2f}, {test_pred[i] + test_IP[i]:.2f}]")
    print("-" * 50)
```

```
Prédiction pour RM = 5 : 10.84
IC à 95%: [9.63, 12.05]
IP à 95%: [-2.21, 23.89]
--------------------------------------------------
Prédiction pour RM = 6 : 19.94
IC à 95%: [19.32, 20.57]
IP à 95%: [6.93, 32.96]
--------------------------------------------------
Prédiction pour RM = 7 : 29.04
IC à 95%: [28.22, 29.87]
IP à 95%: [16.02, 42.07]
--------------------------------------------------
Prédiction pour RM = 8 : 38.15
IC à 95%: [36.62, 39.67]
IP à 95%: [25.06, 51.23]
--------------------------------------------------
```

Rég Simple : Intervalles de Confiance et de Prédiction

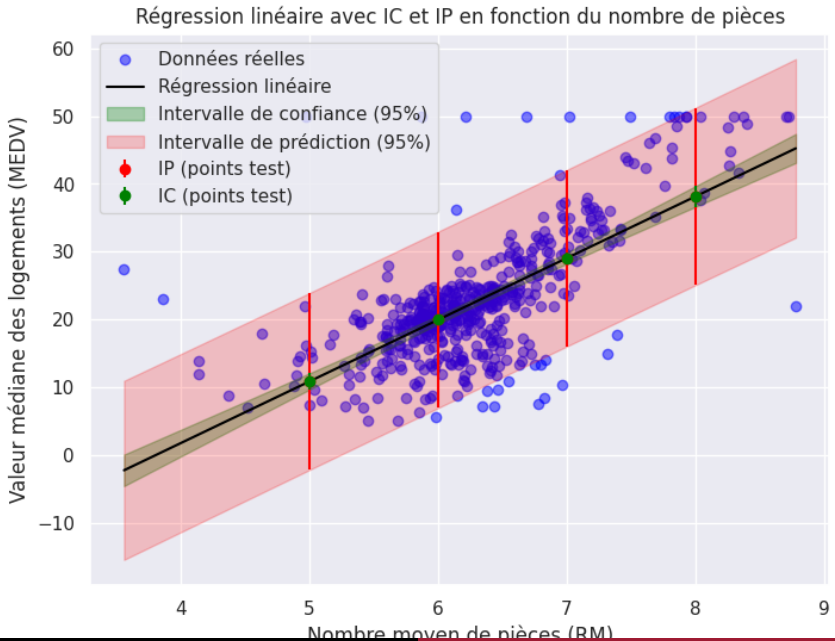
```
# Tracé des intervalles avec marges IC et IP
plt.figure(figsize=(8, 6))
plt.scatter(X['rm'], Y, label="Données réelles", color='blue', alpha=0.5)
plt.plot(rm_values, model.predict(X_range), label="Régression linéaire", color='black')
# Bandes IC et IP
plt.fill_between(rm_values, IC_inf, IC_sup, color='green', alpha=0.3,
                 label="Intervalle de confiance (95%)")
plt.fill_between(rm_values, IP_inf, IP_sup, color='red', alpha=0.2,
                 label="Intervalle de prédiction (95%)")
# Ajout des points de test avec barres d'erreur
plt.errorbar(rm_test_values, test_pred, yerr=test_IP, fmt='o', color='red',
             label="IP (points test)")
plt.errorbar(rm_test_values, test_pred, yerr=test_IC, fmt='o', color='green',
             label="IC (points test)")
plt.xlabel("Nombre moyen de pièces (RM)")
plt.ylabel("Valeur médiane des logements (MEDV)")
plt.legend()
plt.title("Régression linéaire avec IC et IP en fonction du nombre de pièces")
plt.show()
```

Rég Simple : Intervalles de Confiance et de Prédiction



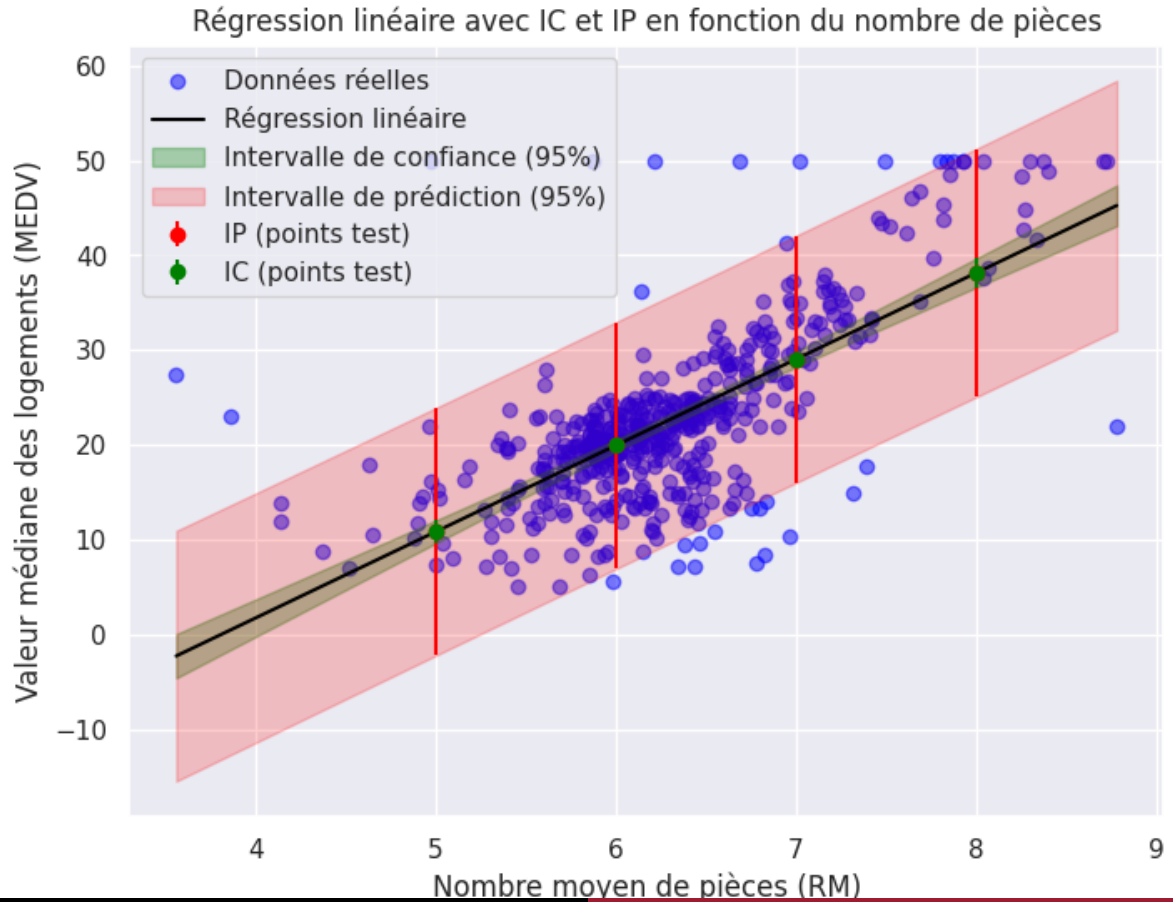
IP et IC : Comment avons-nous obtenu ces figures?

- **Régression Linéaire Simple :** Dans la première figure, nous avons utilisé seulement la variable RM (nombre de pièces) pour prédire la valeur médiane des logements (MEDV). Nous avons effectué les étapes suivantes :
  - Ajustement d'un modèle de régression linéaire simple.
  - Calcul des intervalles de confiance et de prédiction pour chaque valeur de RM.
  - Ajout de points de test (minimum, médiane et maximum de RM) avec leurs propres barres d'erreur pour montrer les différences entre IC et IP.



IP et IC : Comment avons-nous obtenu ces figures?

- **Interprétation de la figure :** La tendance générale montre que plus le nombre de pièces dans un logement est élevé, plus sa valeur médiane a tendance à être haute. L'incertitude est plus faible autour de la médiane de RM (où nous avons plus d'observations) et plus grande aux extrêmes.



Rég Mul : Intervalles de Confiance et de Prédiction

```python
# Importation des bibliothèques nécessaires
import numpy as np
import pandas as pd
import scipy.stats as stats
import matplotlib.pyplot as plt
import statsmodels.api as sm
import seaborn as sns
from ISLP import load_data
sns.set_theme()
# Chargement des données Boston Housing
Boston = load_data('Boston')
# Sélection des variables explicatives et cible
X = Boston.drop(columns=['medv'])
Y = Boston['medv'].values  # Prix médian des logements
# Ajout d'une colonne d'intercept
X_with_intercept = sm.add_constant(X)
# Ajustement du modèle de régression multiple
model = sm.OLS(Y, X_with_intercept).fit()
# Prédictions sur l'échantillon
Y_pred = model.predict(X_with_intercept)
# Calcul des paramètres statistiques
n, p = X.shape
s_squared = np.sum((Y - Y_pred) ** 2) / (n - p - 1)
s = np.sqrt(s_squared)
t_value = stats.t.ppf(1 - 0.025, df=n - p - 1)
# Matrice (X'X)^-1
XTX_inv = np.linalg.inv(X_with_intercept.T @ X_with_intercept)
```

Rég Mul : Intervalles de Confiance et de Prédiction

```
# Définition d'une plage de valeurs pour 'rm'
rm_values = np.linspace(X["rm"].min(), X["rm"].max(), 100)
X_range = np.tile(X.mean(axis=0), (100, 1))
X_range[:, X.columns.get_loc('rm')] = rm_values
# Construire correctement X_range_with_intercept
X_range_df = pd.DataFrame(X_range, columns=X.columns)
# Ajouter l'intercept en s'assurant de l'alignement des colonnes
X_range_with_intercept = np.c_[np.ones(X_range_df.shape[0]), X_range_df]
X_range_with_intercept = pd.DataFrame(
    X_range_with_intercept, columns=X_with_intercept.columns
)
# Prédictions et calcul des marges IC et IP
IC_inf, IC_sup, IP_inf, IP_sup = [], [], [], []
for x_0 in X_range_with_intercept.values:
    x_0 = x_0.reshape(1, -1)  # Assurer une forme correcte (1, p+1)
    variance_IC = s_squared * (x_0 @ XTX_inv @ x_0.T)[0][0]
    variance_IP = s_squared * (1 + x_0 @ XTX_inv @ x_0.T)[0][0]
    SE_IC = np.sqrt(variance_IC)
    SE_IP = np.sqrt(variance_IP)
    y_hat = model.predict(x_0)[0]
    IC_inf.append(y_hat - t_value * SE_IC)
    IC_sup.append(y_hat + t_value * SE_IC)
    IP_inf.append(y_hat - t_value * SE_IP)
    IP_sup.append(y_hat + t_value * SE_IP)
# Définition des points de test pour RM = 5, 6, 7, 8
rm_test_values = np.array([5, 6, 7, 8])
```

Rég Mul : Intervalles de Confiance et de Prédiction

```
# Créer une copie de X avec toutes les variables fixées à leur moyenne
test_X = np.tile(X.mean(axis=0), (len(rm_test_values), 1))
# Modifier uniquement la colonne 'rm' avec les valeurs de test
test_X[:, X.columns.get_loc('rm')] = rm_test_values
# Convertir en DataFrame avec les bonnes colonnes
test_X_df = pd.DataFrame(test_X, columns=X.columns)
# Ajout de l'intercept en s'assurant que les colonnes sont dans le bon ordre
test_X_with_intercept = np.c_[np.ones(test_X_df.shape[0]), test_X_df]
test_X_with_intercept = pd.DataFrame(
    test_X_with_intercept, columns=X_with_intercept.columns
)
# Faire les prédictions
test_pred = model.predict(test_X_with_intercept)
test_IC = []
test_IP = []
for x_0 in test_X_with_intercept.values:
    variance_IC = s_squared * (x_0 @ XTX_inv @ x_0.T)
    variance_IP = s_squared * (1 + x_0 @ XTX_inv @ x_0.T)
    SE_IC = np.sqrt(variance_IC)
    SE_IP = np.sqrt(variance_IP)
    test_IC.append(t_value * SE_IC)
    test_IP.append(t_value * SE_IP)
# Affichage des résultats
for i, rm_val in enumerate(rm_test_values):
    print(f"Prédiction pour RM = {rm_val} : {test_pred[i]:.2f}")
    print(f"IC à 95%: [{test_pred[i] - test_IC[i]:.2f}, {test_pred[i] + test_IC[i]:.2f}]")
    print(f"IP à 95%: [{test_pred[i] - test_IP[i]:.2f}, {test_pred[i] + test_IP[i]:.2f}]")
    print("-" * 50)
```

Rég Mul : Intervalles de Confiance et de Prédiction

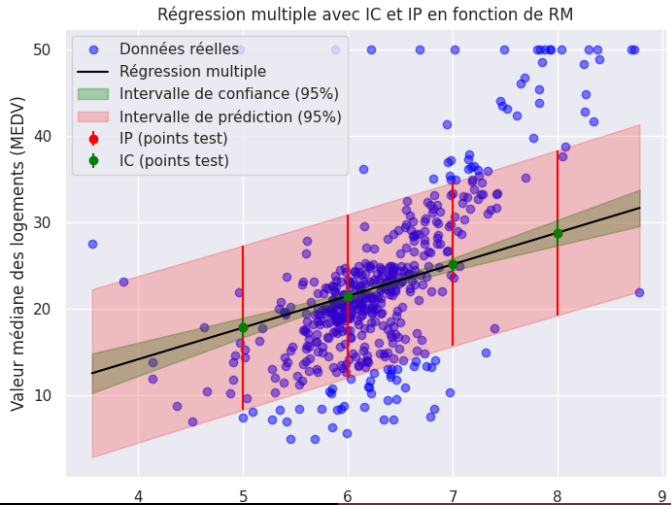
```
# Tracé des intervalles avec marges IC et IP
plt.figure(figsize=(8, 6))
plt.scatter(X["rm"], Y, label="Données réelles", color='blue', alpha=0.5)
plt.plot(rm_values, model.predict(X_range_with_intercept),
         label="Régression multiple", color='black')

# Bandes IC et IP
plt.fill_between(rm_values, IC_inf, IC_sup, color='green', alpha=0.3,
                 label="Intervalle de confiance (95%)")
plt.fill_between(rm_values, IP_inf, IP_sup, color='red', alpha=0.2,
                 label="Intervalle de prédiction (95%)")

# Ajout des points de test avec barres d'erreur
plt.errorbar(rm_test_values, test_pred, yerr=test_IP, fmt='o', color='red',
             label="IP (points test)")
plt.errorbar(rm_test_values, test_pred, yerr=test_IC, fmt='o', color='green',
             label="IC (points test)")
plt.xlabel("Nombre moyen de pièces (RM)")
plt.ylabel("Valeur médiane des logements (MEDV)")
plt.legend()
plt.title("Régression multiple avec IC et IP en fonction de RM")
plt.show()
```

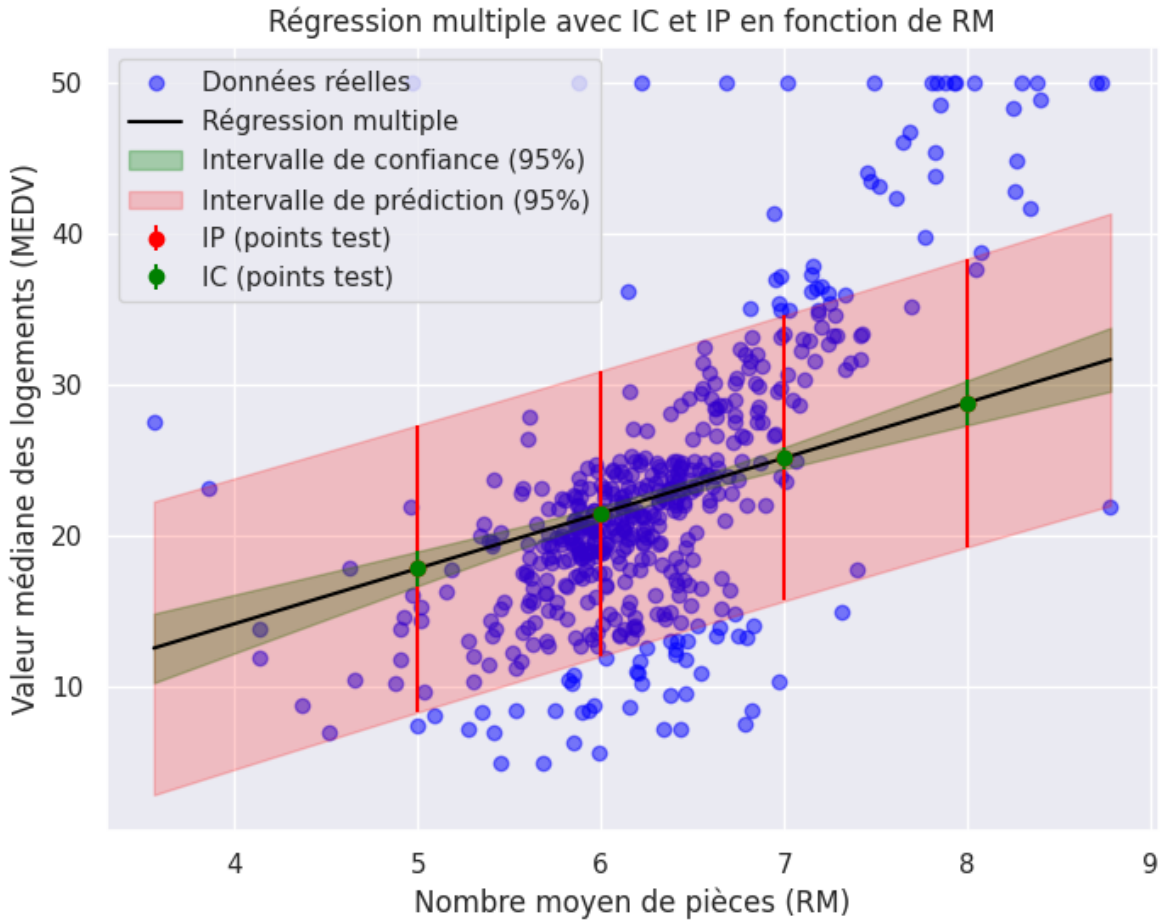
Rég Mul : Intervalles de Confiance et de Prédiction

```
Prédiction pour RM = 5 : 17.83
IC à 95%: [16.69, 18.97]
IP à 95%: [8.34, 27.33]
------------------------------------------------
Prédiction pour RM = 6 : 21.49
IC à 95%: [21.01, 21.97]
IP à 95%: [12.05, 30.93]
------------------------------------------------
Prédiction pour RM = 7 : 25.15
IC à 95%: [24.43, 25.87]
IP à 95%: [15.69, 34.60]
------------------------------------------------
Prédiction pour RM = 8 : 28.81
IC à 95%: [27.33, 30.28]
IP à 95%: [19.27, 38.35]
------------------------------------------------
```



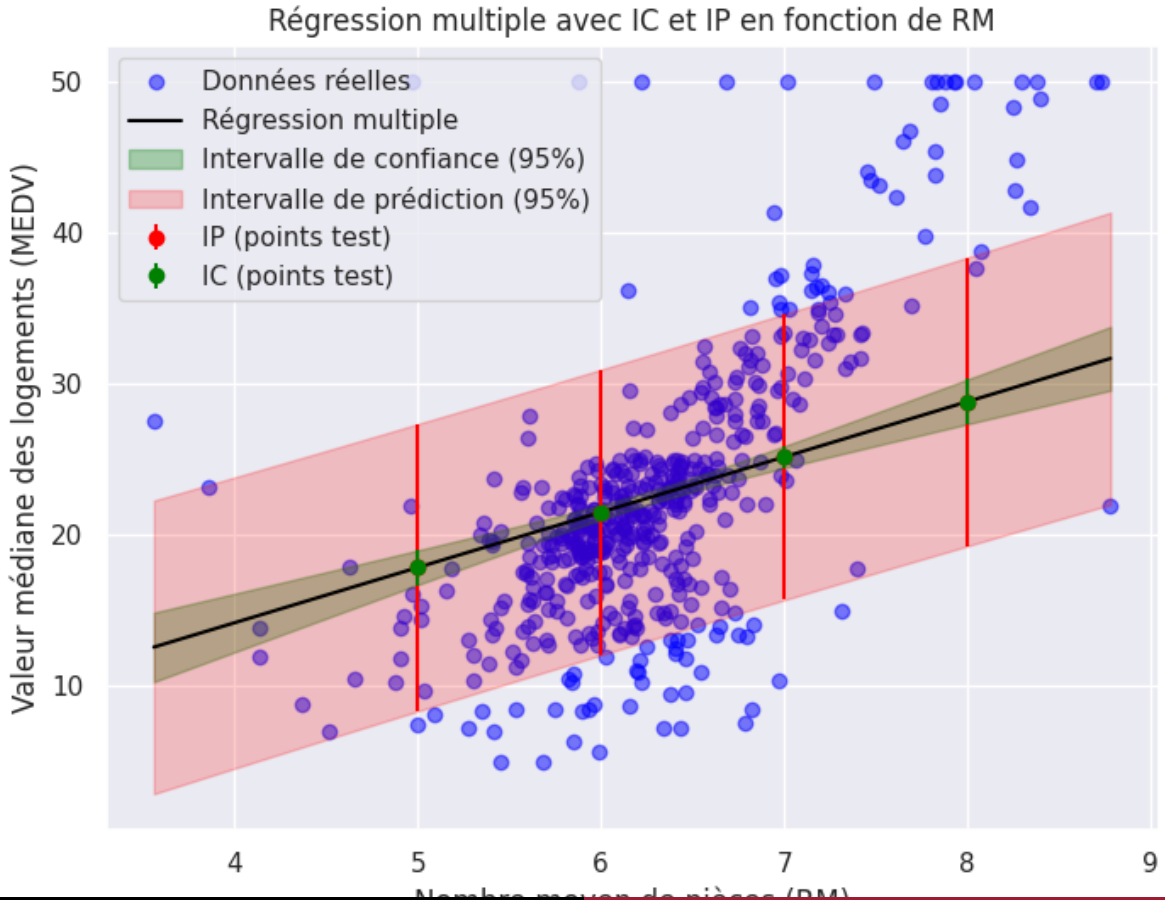
IP et IC : Comment avons-nous obtenu ces figures?

- **Régression Linéaire Multiple :** Dans la seconde figure, nous avons utilisé toutes les variables disponibles pour prédire MEDV, mais nous avons seulement tracé la relation entre RM et MEDV en maintenant toutes les autres variables constantes à leurs moyennes.



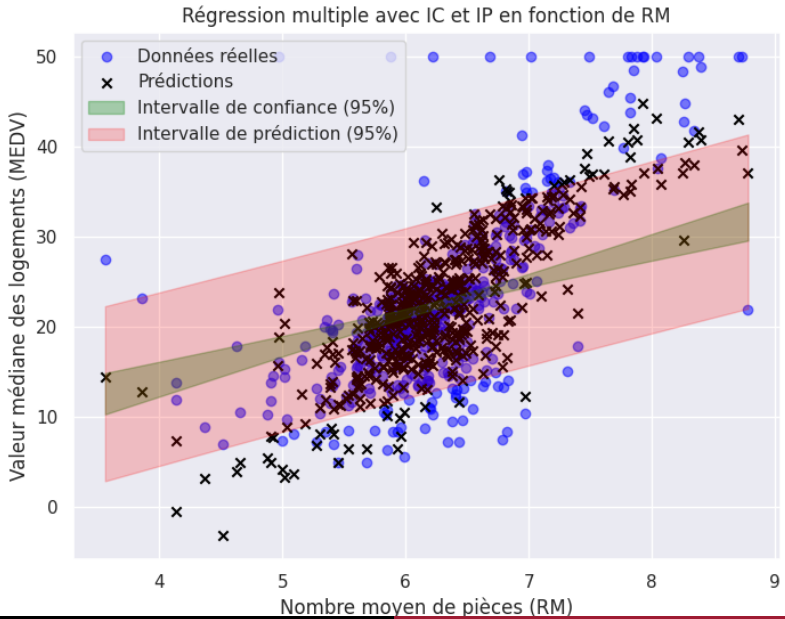
Rég Mul : Intervalles de Confiance et de Prédiction

- **Interprétation de la figure :** On observe une relation similaire entre RM et MEDV, mais avec des intervalles légèrement plus étroits, car l'utilisation de plusieurs variables explicatives améliore la précision du modèle. L'effet du nombre de pièces sur le prix médian du logement est mieux isolé en tenant compte des autres facteurs.



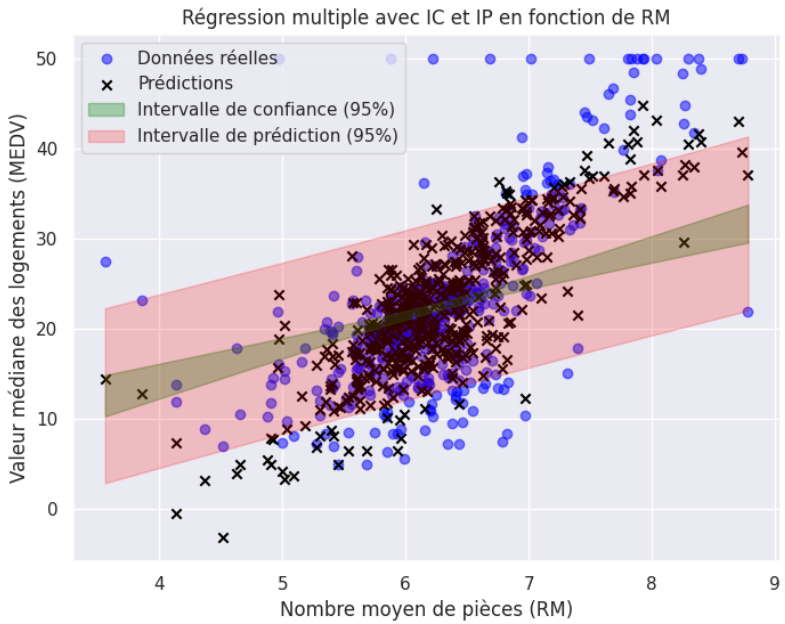
IP et IC : Figure du cours

- **Pourquoi les prédictions sont-elles éparpillées ?** Pendant le cours, on a généré des prédictions (croix noires) ne qui be s'alignent pas parfaitement sur la ligne de tendance de la régression, mais sont éparpillées pour plusieurs raisons :
- **Le modèle est une régression multiple :**
  - Contrairement à une régression simple, où la variable RM expliquerait à elle seule MEDV, ici le modèle prend en compte toutes les variables explicatives disponibles dans le dataset.
  - Les prédictions de MEDV pour une même valeur de RM varient, car les autres variables influencent aussi les résultats.



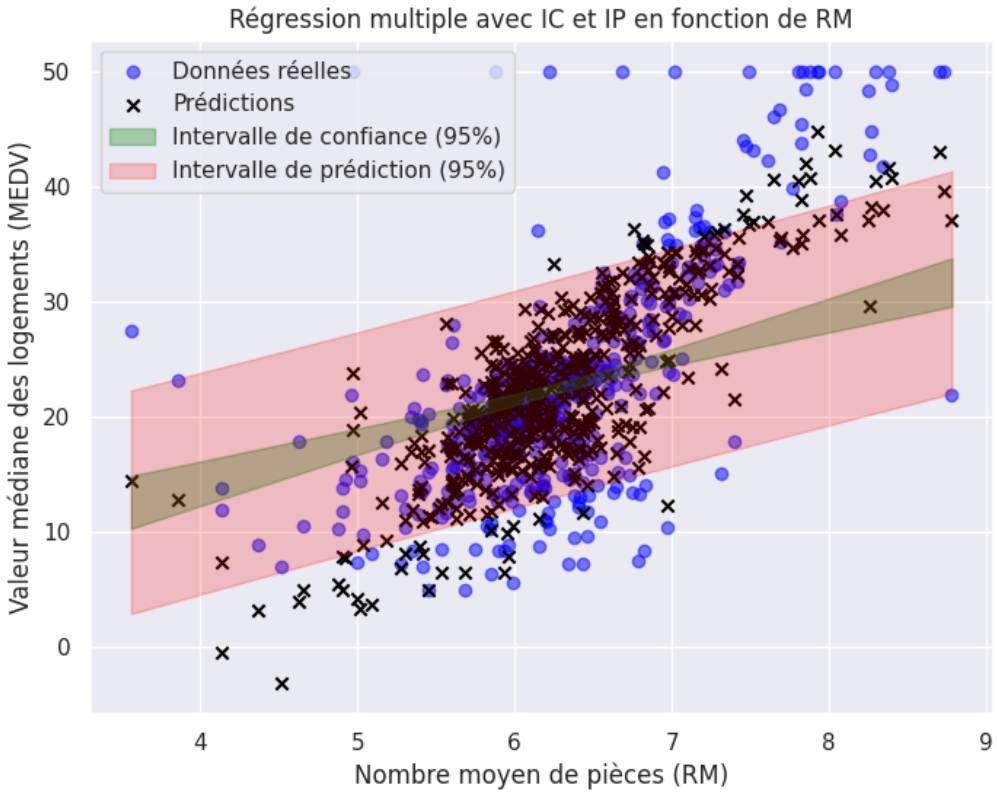
IP et IC : Figure du cours

- **Le graphe représente une projection d'un modèle multidimensionnel :**
  - Le modèle de régression multiple utilise plusieurs variables (ex. taux de criminalité, accès aux autoroutes, âge des logements, etc.).
  - Dans ce graphique, nous ne visualisons qu'une seule variable (RM), ce qui donne l'impression que les prédictions sont éparpillées.
  - Chaque croix noire correspond à une prédiction fixant toutes les autres variables explicatives à leurs valeurs réelles (au lieu de les fixer à leur moyenne, comme dans la courbe de tendance).



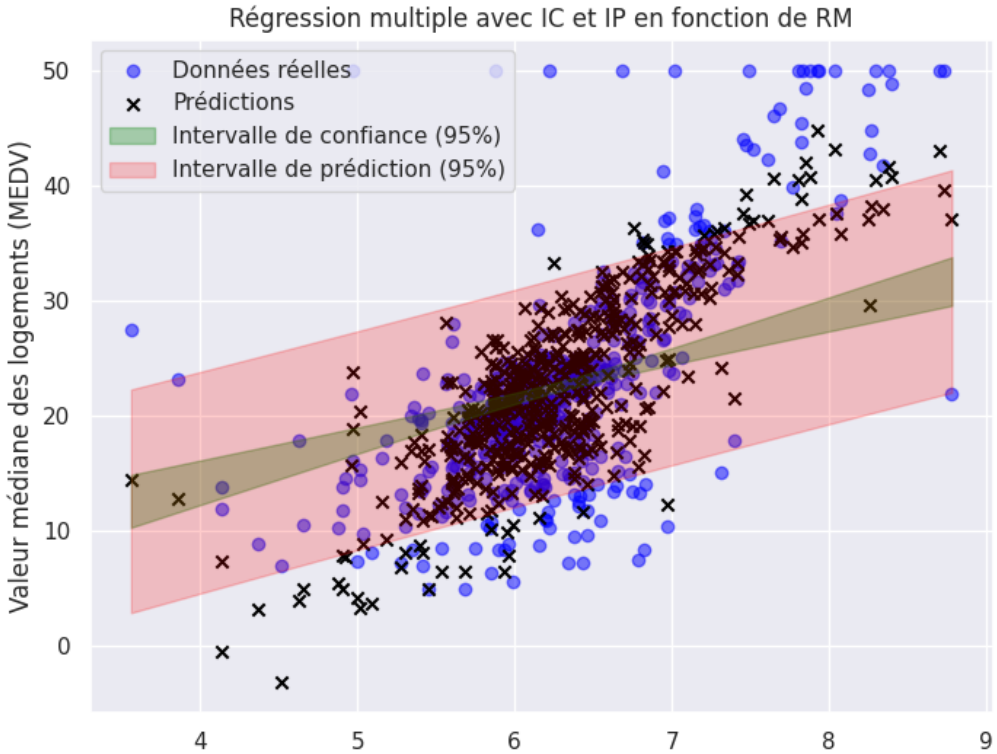
IP et IC : Figure du cours

- **Les autres variables influencent fortement la prédiction :**
  - Deux logements avec le même RM peuvent avoir des valeurs de MEDV très différentes en raison d'autres facteurs (ex. localisation, pollution, etc.).
  - Par conséquent, pour une valeur donnée de RM, les prédictions (croix noires) se répartissent sur une large plage de valeurs de MEDV, expliquant cette dispersion.



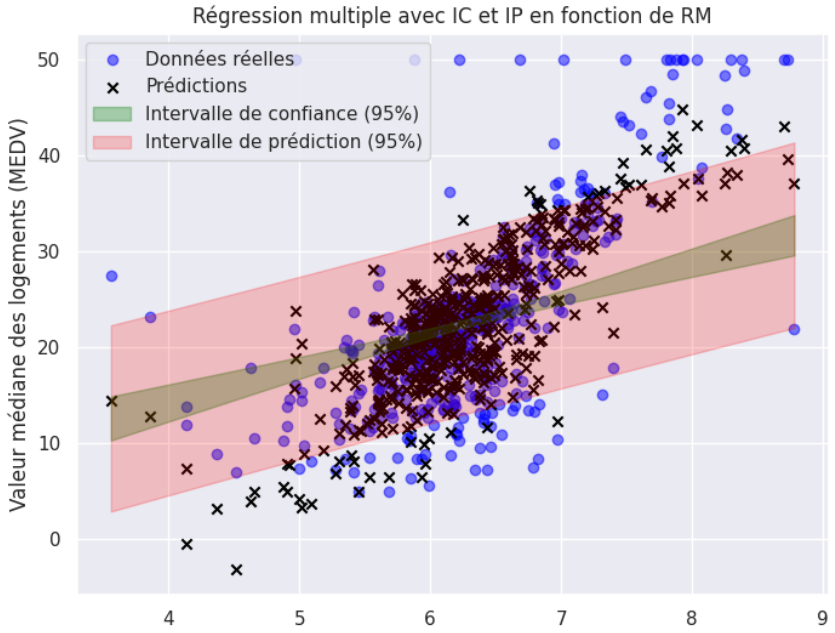
IP et IC : Figure du cours

- **Les intervalles de confiance et de prédiction restent structurés:**
  - L'intervalle de confiance (IC) représente la certitude sur la moyenne estimée de MEDV pour un RM donné.
  - L'intervalle de prédiction (IP) reflète la variabilité des données, c'est pourquoi il est beaucoup plus large.
  - Même si les prédictions sont dispersées, elles restent majoritairement dans les limites de l'IP.



Conclusion : Projection Réduite d'un Modèle Complexe

- Le graphe illustre une relation simplifiée entre MEDV et RM, mais en réalité, la valeur des logements dépend de plusieurs facteurs simultanément.
- L'éparpillement des prédictions indique que le nombre moyen de pièces (RM) ne suffit pas à expliquer à lui seul la valeur des logements.
- La dispersion des prédictions n'est pas une erreur, mais reflète simplement la complexité du modèle et l'influence des autres variables explicatives.



IC et IP : Résumé

- **IC** : Intervalle de confiance $\rightarrow$ Incertitude sur la réponse moyenne estimée pour une observation donné.
- **IP** : Intervalle de prédiction $\rightarrow$ Incertitude sur une nouvelle valeur de la réponse pour une observation donnée.
- **Régression linéaire simple** : Relation entre la variable explicative X et la réponse Y en ignorant les autres variables.
- **Régression linéaire multiple** : Relation entre une variable explicative X_i et la réponse Y en tenant compte de toutes les autres variables X_j.
- **À retenir :** Dans les deux cas, la largeur des intervalles augmente où il y a moins de données et plus d'incertitude.