1 Régression Linéaire simple et moindres carrés

1.1 Régression Linéaire simple et moindres carrés

1.1.1 Régression Linéaire simple et moindres carrés

Considérons le modèle de régression linéaire simple

$$Y_i = \beta_0 + \beta_1 x_i + E_i, \tag{1}$$

les erreurs E_i sont supposées centrées et dé-corrélées. Soit l'échantillon suivant qui représente le pourcentage de rendement, y_i, en fonction de la température x_i en °C d'un procédé chimique

```
i :  1  2  3  4  5  6  7  8  9  10
x :  45 50 55 60 65 70 75 80 85 90
y  : 43 45 48 51 55 57 59 63 66 68
```

Il est prévu que le pourcentage du rendement d'un procédé chimique est liée linéairement à la température. L'objectif est donc d'estimer un modèle de régression linéaire simple à partir de cet échantillon. Pour cela :

1. Créez une fonction qui calcule et renvoie les cœfficients de régression à partir des données. Pour cela, vous commencerez par la utiliser les deux premières formules vues en cours pour estimer les deux paramètres d'un modèle de régression linaire :
   – Calculez $\bar{x}$ et $\bar{y}$,
   – En déduire les valeurs des cœfficients de régression par moindres carrés.
2. Représentez graphiquement les données et la droite de régression obtenue (sur le même graphique)
3. Donnez une prédiction de la valeur de y pour $x = 100$ degrés. Justifier votre prédiction
4. Comparez vos résultats avec ceux que donne la fonction `polyfit` de Matlab

1.1.2 Formulation vectorielle

Maintenant considérons le même modèle sous sa formulation vectorielle

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e} \tag{2}$$

où $\boldsymbol{\beta} = (\beta_0, \beta_1)^T$ est le vecteur des coefficients de régressions inconnus à estimer, $\mathbf{X}$ la matrice de régression (ou de design), $\mathbf{y}$ le vecteur des données des sorties y_i et $\mathbf{e}$ le vecteur des erreurs inconnues. Afin d'estimer le vecteur paramètre $\boldsymbol{\beta}$ du modèle, vous allez créer une fonction qui calcule et renvoie ce vecteur des coefficients de régression à partir des données :

1. Dans cette fonction
   – construisez le vecteur $\mathbf{y} = (y_1, \ldots, y_n)^T$ des valeurs d'observations de Y et la matrice de design $\mathbf{X}$
   – Implémentez la formule matricielle des MC pour estimer $\boldsymbol{\beta}$
2. Représentez graphiquement les données et la droite de régression obtenue (sur le même graphique)
3. Donnez une prédiction de la valeur de y pour $x = 100$ degrés. Justifier votre prédiction
4. Testez graphiquement la normalité des errurs en utilisant la fonction Matalab `qqplot`
5. Comparez vos résultats avec ceux que donne la fonction `polyfit` de Matlab

2 Régression polynomiale simple, moindres carrés et tests

Maintenant considérons le modèle de régression polynomiale où l'objectif est d'estimer un polynôme de degré p plutôt qu'une simple droite. Le modèle est donc formulé ainsi

$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots + \beta_p x_i^p + E_i, \tag{3}$$

Vous allez donc créer une fonction qui permet d'estimer les coefficients $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^T$ d'une fonction de régression polynomiale de degré p à partir d'un échantillon. Pour cela :

1. Créez une nouvelle fonction qui calcule et renvoie le vecteur des coefficients de régression à partir des données, pour un degrés de polynôme p donné.
2. Représentez graphiquement les données et la courbe de régression obtenue (sur le même graphique) pour le cas où $p = 2$
3. Donnez une prédiction de la valeur de y pour $x = 100$ degrés. Justifier votre prédiction
4. Testez graphiquement la normalité des errurs en utilisant la fonction Matalab `qqplot`
5. Comparez vos résultats avec ceux que donne la fonction `polyfit` de Matlab

2.1 Régression linéaire et descente de gradient

Ici il s'agit d'utiliser l'algorithme de gradient pour estimer les paramètres de régression (bien que l'on ait une solution exacte, l'idée ici est de voir la notion d'optimisation globale et méthode itérative ...)

Pour rappel, l'algorithme de gradient est comme suit. On se donne un paramètre initial $\boldsymbol{\beta}0$ et un seuil de tolérance $\epsilon \geqslant 0$ (pour le teste de convergence).

– Initialisation : $\boldsymbol{\beta} = \boldsymbol{\beta}^0$
  Répéter
  $\boldsymbol{\beta}^t = \boldsymbol{\beta}^{t-1} - \lambda \frac{\partial f(\boldsymbol{\beta})}{\boldsymbol{\beta}}$
  tan que $\| \boldsymbol{\beta}^t - \boldsymbol{\beta}^{t-1} \| < \epsilon$

λ étant le pas de descente $\in [0, 1]$

1. Créez une fonction implémentant l'algorithme de gradient pour la régression polynomiale
2. Considérez un des jeux de données précédents et comparez les résultats obtenus par la méthode de gradient avec le cas de la minimisation directe des moindres carrés
3. Faites cela sur un graphique (affichage des résultats (droite de régression dans le cas linéaire simple))
4. Considérez le cas de la régression linéaire simple et représenter graphiquement le critère de somme des erreurs quadratiques en fonction de β_0 et β_1
5. Que remarquer vous ?
6. Représentez graphiquement (sur le même graphique) le résultat obtenu après estimation par moindres carrés de β_0 et β_1
7. que remarquer vous ?

2.2 Intervalle de Confiance : Loi Normale univariée $\mathcal{N}(x; \mu, \sigma^2)$

1. Créez une fonction qui calcule et renvoie l'intervalle de confiance à 95% sur l'espérance d'une loi normale

   Votre fonction devra prendre en compte le fait si la variance σ^2 est connue ou pas
2. Une fonction qui calcule et renvoie l'intervalle de confiance à 95% sur la variance d'une loi normale

   Votre fonction devra prendre en compte le fait si l'espérance μ est connue ou pas
3. Simulez un échantillon Gaussien *i.i.d.* suivant la densité $\mathcal{N}(x; \mu, \sigma^2)$ avec des paramètres de votre choix
4. Tester votre fonction

3 régression linéaire (jeux de données bis)

Pour estimer le volume en bois utilisable d'un arbre debout en fonction de l'aire du tronc mesuré à 25 cm du sol. On a choisi au hasard 10 arbres et mesuré à la base, l'aire correspondante (en cm^2). On a ensuite enregistré, une fois l'arbre coupé, le volume correspondant en m^3 .

```
Vol :  0,152  0,284  0,187  0,350  0,416  0,230  0,242  0,276  0,383  0,140
Aire :  297  595  372  687  790  520  473  585  762  232
```

Faites le même travail

un autre jeu de données

```
x_i : 18 7  14 31 21 5  11 16 26 29
y_i : 55 17 36 85 62 18 33 41 63 87
```

Enjoy !