

Modèles de Prédiction de la température annuelle Moyenne en France

Travail Encadre de Recherche

Axel POTIER et Cheickna DIABY encadré par Azzouz DERMOUNE

Année 2018

Table des matières

1	Méthode d'estimation de la température par une famille d'estimateurs	4
1.1	Analyse des Données	4
1.2	Premier modèle de prévision : famille de prévisions et choix optimal	5
2	Modèle de la marche Aléatoire	9
2.1	Estimation de m_I et de la variance σ_I^2	9
2.2	Tendance linéaire	11
2.3	Intervalle de confiance	12
2.4	Loi du vecteur (S_0, \dots, S_n)	12
2.5	Le cas des covariances nulles	13
2.6	Cas général	13
2.7	Algorithme d'Anderson	17
2.8	Cas particulier	19
2.8.1	Quelques propriétés	19
2.8.2	Application de l'algorithme d'anderson	20
3	Arbre de régression	23
3.1	Théorie sur les arbres de régression	23
3.1.1	Algorithme de prédiction par un modèle à 2 feuilles	25
A	Le code	30
A.1	Graphe des Températures moyennes en fonction de la date	30
A.2	Estimation modèle 1	30
A.3	Estimation de s_{116}	32
A.4	test de gaussiannité des résidus	32

B	Modèle de la Marche aléatoire	33
B.1	Test de gaussianité des incréments	33
B.2	estimation des paramètres m_i et σ_I^2	33
B.3	Intervalle de Confiance de s_{n+1}	34
B.4	Estimation de Tendance	34
B.5	Algorithme d'Anderson	35
C	Arbre de Régression	38
C.1	Code arbre à deux feuilles	38
C.2	estimation de s_{n+1} avec le modèle de l'arbre à deux feuilles	39
C.3	Code de l'arbre de régression	40
	Bibliographie	41

Introduction

Au cours de ce TER, on propose trois modèles de prédiction de la température moyenne de l'année 2016, sur la base des températures moyennes de 1900 à 2015. Le premier modèle est le choix optimal parmi une famille d'estimateurs : chacun prenant la moyenne sur les températures des k dernières années, et ceci pour $k = 1, \dots, 116$. Le deuxième modèle est celui de la marche aléatoire. Le dernier modèle est un arbre de régression. En annexe on peut retrouver le tableau de données.

Partie 1

Méthode d'estimation de la température par une famille d'estimateurs

1.1 Analyse des Données

Le tableau de données sur lequel on travail possède une colonne : les températures annuelles moyennes entre les années 1900 à 2015. Cette colonne est représentée par $(s_0, \dots, s_n)^\top$, où $n + 1 = 116$ est le nombre des observations, c'est-à-dire s_i désigne la température annuelle moyenne de l'année $1900 + i$. La série temporelle $i \rightarrow s_i$ est représentée dans la figure 1.1. La moyenne empirique des températures de 1900 à 2015 est 11.97 et la variance vaut 0.465531.

On observe deux tendances, avant 1980 on a un groupe qui semble être une réalisation d'une suite de variables aléatoires i.i.d., et après 1980 le graphique affiche une pente. Ceci implique un changement de la loi de probabilité à partir de 1980. Cette observation montre que la moyenne empirique 11.97 ne peut pas être une bonne prévision de la température annuelle moyenne de l'année 2016.

Le code utilisé est en annexe A.1.

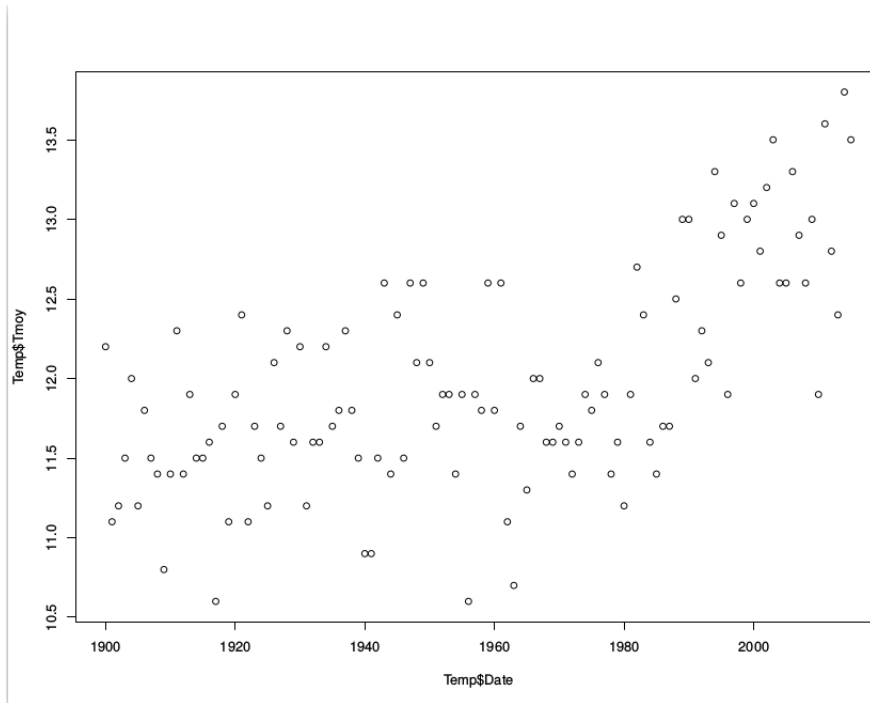


FIGURE 1.1 – Température moyenne de 1900 à 2015

1.2 Premier modèle de prévision : famille de prévisions et choix optimal

L'objectif du premier algorithme est de prédire la température de l'année $n + 1$ par le meilleur estimateur dans la famille

$$\hat{s}_{n+1}(k) := \frac{\sum_{i=n+1-k}^n s_i}{k} \quad (1.1)$$

(la moyenne empirique des k derniers termes de la suite s_0, \dots, s_n), avec $k = 1, \dots, n + 1$. Par exemple pour $k = n + 1$, on obtient l'estimateur par la moyenne empirique $\bar{s} = 11.97155$. Quel est le meilleur choix de k ?

Si on connaît la valeur de s_{n+1} , alors le meilleur choix de k est

$$\arg \min \{e(n + 1, k) := |s_{n+1} - \frac{\sum_{i=n+1-k}^n s_i}{k}| : k = 1, \dots, n + 1\} \quad (1.2)$$

En revanche on peut calculer les erreurs

$$e(l+1, k) = \left| s_{l+1} - \frac{\sum_{i=l+1-k}^l s_i}{k} \right|, \quad l < n, k \leq l+1, \quad (1.3)$$

$$e(l+1, \min(k, l+1)), \quad k = 1, \dots, n, l = 0, \dots, n-1, \quad (1.4)$$

$$E(k) = \max\{e(l+1, \min(k, l+1)), \quad l = 0, \dots, n-1\}, \quad k = 1, \dots, n+1. \quad (1.5)$$

Le meilleur k est celui qui minimise $k = 1, \dots, n \rightarrow E(k)$. On observe le graphique suivant :

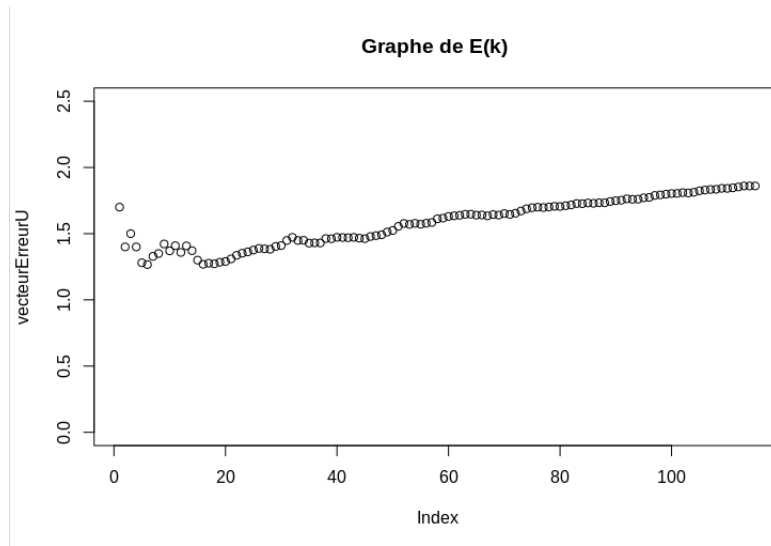


FIGURE 1.2 – Graphe de $E(k)$

Le meilleur estimateur est celui pour $k = 6$, c'est-à-dire la meilleure prévision de s_{l+1} est donnée par $\hat{s}_{l+1} = \sum_{i=1}^6 \frac{s_{l+1-i}}{6}$. Donc la meilleure prévision de 2016 est :

$$\hat{s}_{n+1} = \sum_{i=1}^6 \frac{s_{n+1-i}}{6} = 12.91667$$

Le code est dans l'annexe A.3. L'analyse des résidus $e_l = s_{l+1} - \hat{s}_{l+1}$ est donnée dans le graphique 1.4. Le code est en annexe A.2.

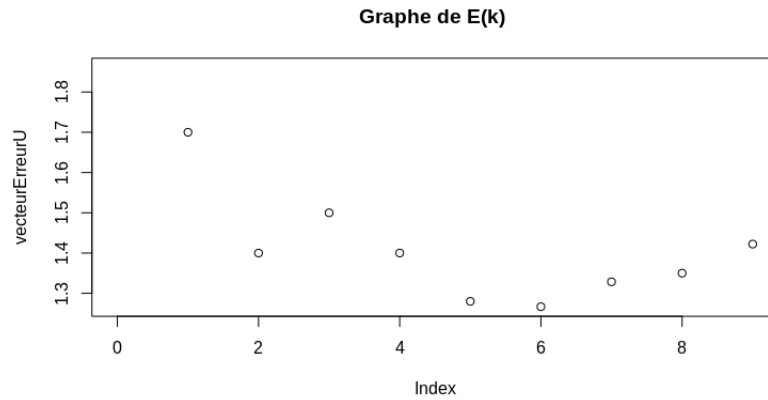
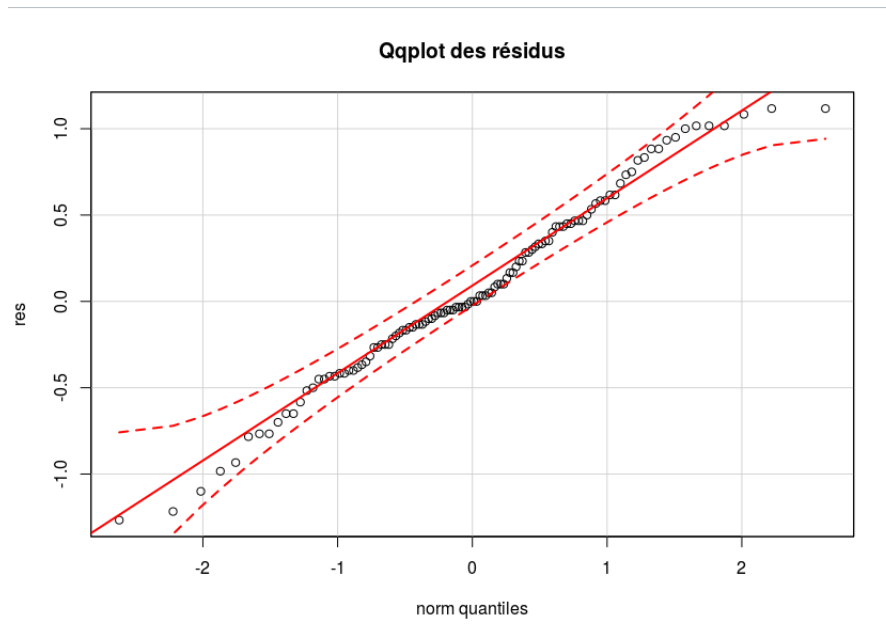


FIGURE 1.3 – Zoom vers l’optimum de E(k)

Sur le Qqplot des résidus e_1, \dots, e_n , on observe qu’ils se comportent comme un échantillon i.i.d de loi gaussienne. on peut estimer les paramètres $\hat{m}_{res} = 0.06184211$ et de variance $\sigma_{res}^2 = 0.2787204$. On peut réécrire du coup : $s_{l+1} = \sum_{i=1}^6 \frac{s_{l+1-i}}{6} + e_{l+1}$ avec e_l qui suit $\mathcal{N}(m_{res}, \sigma_{res}^2)$.



H

FIGURE 1.4 –

Partie 2

Modèle de la marche Aléatoire

Définition 2.1. Soit $S_i = S_0 + X_1 + \dots + X_i$, avec $i = 1, \dots, n$ est une marche aléatoire si les accroissements $(X_1 = S_1 - S_0, \dots, X_n = S_n - S_{n-1})$ sont des v.a iid.

Remarque Le q-q plot 2.1 des accroissements $s_1 - s_0, \dots, s_n - s_{n-1}$ de la température annuelles moyennes est celui de n réalisations i.i.d. d'une loi gaussienne de moyenne m_I et de variance σ_I^2 . Donc s_0, \dots, s_n est une réalisation d'une marche aléatoire S_0, \dots, S_n . La loi commune des accroissements $S_i - S_{i-1}$ est gaussienne de moyenne m_I et de variance σ_I^2 .

2.1 Estimation de m_I et de la variance σ_I^2

Nous avons n réalisations $s_1 - s_0, \dots, s_n - s_{n-1}$ de la loi gaussienne de moyenne m_I et de variance σ_I^2 . Donc la moyenne m_I est estimée par

$$\hat{m}_I = \frac{\sum_{i=1}^n (s_i - s_{i-1})}{n} = \frac{s_n - s_0}{n} = 0.0112069$$

et la variance σ_I^2 est estimée par

$$\hat{\sigma}_I^2 = \frac{\sum_{i=1}^n (s_i - s_{i-1} - \hat{m}_I)^2}{n} = 0.437738$$

Proposition 2.1. *Comme nous sommes dans le cas gaussiens, alors ces estimateurs sont les estimateurs du maximum de vraisemblance.*

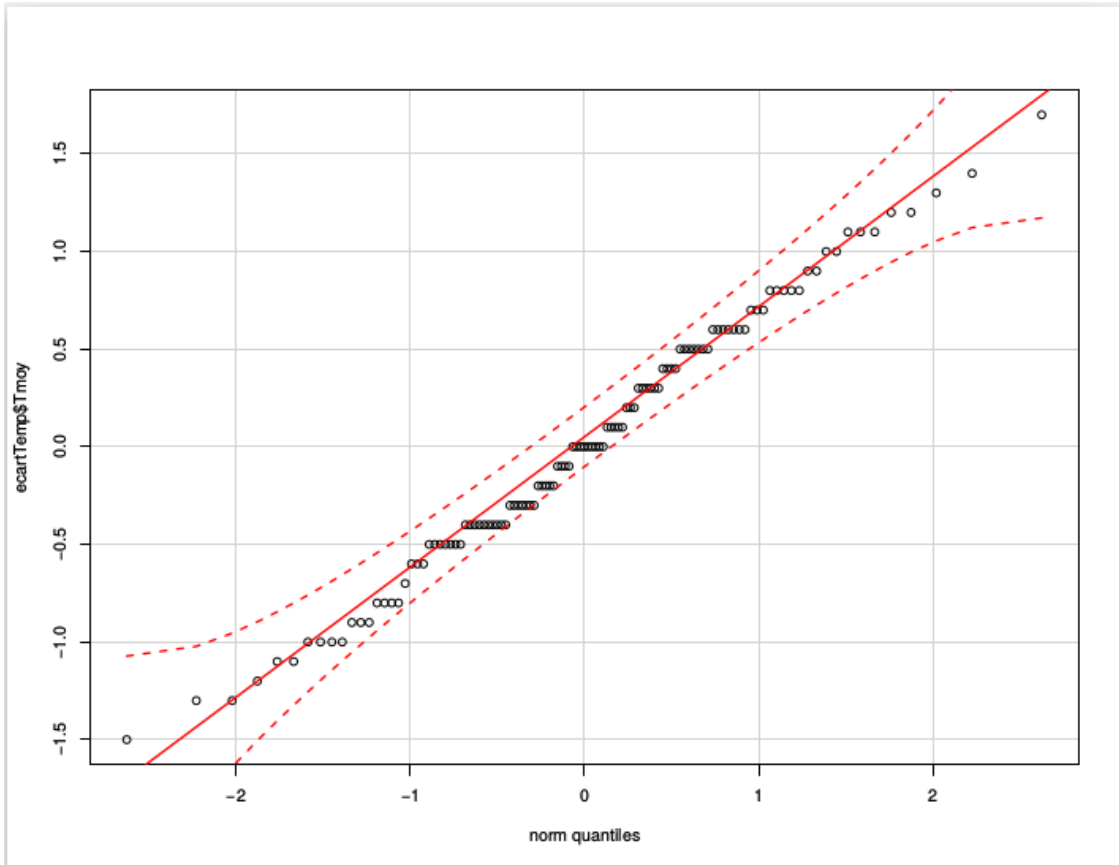


FIGURE 2.1 – Qqplot des Accroissements

Preuve 1. On cherche m_I et σ_I^2 tel que

$$\partial_{m_I} l(s_1 - s_0, \dots, s_n - s_{n-1}, m_I, \sigma_I^2) = 0 \quad (2.1)$$

et tel que

$$\partial_{\sigma_I^2} l(s_1 - s_0, \dots, s_n - s_{n-1}, m_I, \sigma_I^2) = 0 \quad (2.2)$$

On obtient pour la première des équations :

$$(-2) \sum_{i=1}^n \frac{(s_i - s_{i-1} - m_I)}{\sigma_I^2} = 0. \quad (2.3)$$

Soit :

$$\sum_{i=1}^n (s_i - s_{i-1} - m_I) = 0. \quad (2.4)$$

donc

$$\sum_{i=1}^n (s_i - s_{i-1}) = n \hat{m}_I \quad (2.5)$$

donc $m_I = \frac{s_n - s_0}{n}$.

Pour le calcul de σ_I^2 ,

$$\partial_{\sigma_I^2} l(s_1 - s_0, \dots, s_n - s_{n-1}, m_I, \sigma_I^2) = 0 \iff \quad (2.6)$$

$$\partial_{\sigma_I^2} (n \ln \sigma_I^2) + \sum_{i=1}^n \frac{(s_i - s_{i-1} - m_I)^2}{\sigma_I^2} = 0 \quad (2.7)$$

$$\iff \frac{n}{\sigma_I^2} = \sum_{i=1}^n \frac{(s_i - s_{i-1} - m_I)^2}{\sigma_I^4} \quad (2.8)$$

$$\iff \sigma_I^2 = \frac{1}{n} \sum_{i=1}^n (s_i - s_{i-1} - m_I)^2 \quad (2.9)$$

On pose $\hat{\sigma}_I^2 = \frac{1}{n} \sum_{i=1}^n (s_i - s_{i-1} - m_I)^2$

2.2 Tendence linéaire

Pour calculer la moyenne de S_i on utilise l'égalité

$$S_i = S_0 + (S_1 - S_0) + \dots + (S_i - S_{i-1}).$$

D'où

$$\mathbf{E}[S_i] = \mathbf{E}(S_0) + i m_I = m_0 + i m_I.$$

Par conséquent la trajectoire moyenne $i \rightarrow \mathbf{E}(S_i)$ est une droite (tendance linéaire).

Sa pente est estimée à

$$\hat{m}_I = 0.0112069$$

En revanche son intercept m_0 est inconnue.

2.3 Intervalle de confiance

Le modèle nous dit que $S_{n+1} - S_n$ est gaussienne de moyenne m_I et de variance σ_I^2 . Donc le rapport $\frac{S_{n+1} - S_n - m_I}{\sigma_I}$ suit la loi normale standard. Par conséquent avec la probabilité 0.95 l'observation $\frac{s_{n+1} - s_n - m_I}{\sigma_I}$ se trouve dans l'intervalle $[-1.96, 1.96]$. D'où

$$s_{n+1} \in [s_n + m_I - 1.96\sigma_I, s_n + m_I + 1.96\sigma_I] = [12.214437, 14.807977]$$

avec la probabilité 0.95.

En utilisant la tendance, la prévision de l'année $n + 1$ est égale à

$$m_0 + (n + 1)m_I.$$

En imposant que $m_0 + (n + 1)m_I \in [s_n + m_I - 1.96\sigma_I, s_n + m_I + 1.96\sigma_I]$ et donc on a que avec probabilité 0.95

$$m_0 \in [10.9032297, 13.4967697]$$

Remarque. A titre informatif la prédiction du modèle 1, $\hat{s}_{n+1}(6) = 12.91667$ se trouve dans cet intervalle de confiance.

2.4 Loi du vecteur (S_0, \dots, S_n)

Pour connaître cette loi nous avons besoin des covariances

$$r_i = \text{cov}(S_0, S_i - S_{i-1}), \quad i = 1, \dots, n,$$

et de la variance σ_0^2 de S_0 . Sachant ces paramètres la matrice de covariance de $(S_0, S_1 - S_0, \dots, S_n - S_{n-1})$ a pour forme

$$C = \begin{pmatrix} \sigma_0^2 & r_1 & \dots & \dots & \dots & r_n \\ r_1 & \sigma_I^2 & 0 & \dots & \dots & 0 \\ \vdots & 0 & \sigma_I^2 & 0 & \dots & 0 \\ \vdots & \dots & \ddots & \ddots & \ddots & \vdots \\ r_n & 0 & \dots & \dots & \dots & \sigma_I^2 \end{pmatrix} \quad (2.10)$$

La $-2 \log$ de la vraisemblance du vecteur $(S_0, S_1 - S_0, \dots, S_n - S_{n-1})$ est égale à

$$l(m_0, \sigma_0^2, r_1, \dots, r_n) = (n + 1) \ln(2\pi \det(C)) + (s_0 - m_0, s_1 - s_0 - m_I, \dots, s_n - s_{n-1} - m_I) C^{-1} (s_0 - m_0, s_1 - s_0 - m_I, \dots, s_n - s_{n-1} - m_I)^\top.$$

2.5 Le cas des covariances nulles

La $-2 \log$ de la vraisemblance du vecteur $(S_0, S_1 - S_0, \dots, S_n - S_{n-1})$ sous $r_1 = \dots = r_n = 0$ devient

$$l(m_0, \sigma_0^2) = (n+1) \ln(2\pi\sigma_0^2) + (n+1)n \ln(2\pi\sigma_I^2) + \frac{(s_0 - m_0)^2}{\sigma_0^2} + \sum_{i=1}^n \frac{(s_i - s_{i-1} - m_I)^2}{\sigma_I^2}.$$

Proposition 2.2. *Les estimateurs de vraisemblance de m_0 et σ_0^2 sont respectivement*

$$m_0 = s_0, \quad \hat{\sigma}_0^2 = 0$$

Le ML nous dit qu'on doit considérer S_0 comme une constante. Donc (S_0, \dots, S_n) est un mouvement brownien discret. Sachant S_n , S_{n+1} est indépendant du passé.

2.6 Cas général

Proposition 2.3. *Il existe un unique vecteur ligne (w_0, \dots, w_n) telles que*

$$\sum_{i=0}^n w_i = 1$$

et $(\sum_{i=0}^n w_i S_i, S_1 - S_0, \dots, S_n - S_{n-1})$ soient indépendantes.

Proposition 2.4 (Dermoune et al.). *1) Pour chaque vecteur ligne (w_0, \dots, w_n) avec*

$$\sum_{i=0}^n w_i = 1$$

et pour chaque $j = 1, \dots, n$ nous avons

$$\text{cov}(\sum_{i=0}^n w_i S_i, S_j - S_{j-1}) = r_j + \sigma_I^2 \sum_{i=j}^n w_i.$$

2) *Les variables aléatoires $\sum_{i=0}^n w_i S_i, S_1 - S_0, \dots, S_n - S_{n-1}$ sont indépendantes si et seulement si*

$$w_n = -\frac{r_n}{\sigma_I^2} \tag{2.11}$$

$$w_j = \frac{r_{j+1} - r_j}{\sigma_I^2}, \quad j = 1, \dots, n-1, \tag{2.12}$$

$$w_0 = 1 + \frac{r_1}{\sigma_I^2}. \tag{2.13}$$

3) On note $S^{(n)} = \sum_{i=0}^n w_i S_i$. Soit \mathbf{w} est tel que les variables aléatoires $\mathbf{w}S^{(n)}$, $S_1 - S_0, \dots, S_n - S_{n-1}$ soient indépendantes. La variance de $\sum_{i=0}^n w_i S_i$ est égale à

$$\sigma_w^2 = \sigma_0^2 - \frac{\sum_{i=1}^n r_i^2}{\sigma_I^2}. \quad (2.14)$$

Démonstration. preuve de 1) : $cov(\sum_{i=0}^n w_i S_i, S_j - S_{j-1}) = cov(\sum_{i=0}^n w_i S_i, S_j - S_{j-1})$

$$\begin{aligned} &= \sum_{i=0}^n w_i cov(S_i, S_j - S_{j-1}) \\ &= w_0 cov(S_0, S_j - S_{j-1}) + \sum_{i=1}^n w_i cov(S_0 + \sum_{k=1}^i (S_k - S_{k-1}), S_j - S_{j-1}) \\ &= w_0 r_j + cov(S_0, S_j - S_{j-1}) \sum_{i=1}^n w_i + \sum_{i=1}^n \sum_{j=1}^i w_i cov(S_j - S_{j-1}, S_i - S_{i-1}) \\ &= r_j + \sigma_I^2 \sum_{i=j}^n w_i. \end{aligned}$$

preuve de 2) immédiat avec le résultat 1) soit : $r_j + \sigma_I^2 \sum_{i=j}^n w_i = 0$.

Preuve de l'assertion 3) On a

$$var(\mathbf{w}S^{(n)}) = cov(\mathbf{w}S^{(n)}, \sum_{i=0}^n w_i S_i) = w_0 cov(\mathbf{w}S^{(n)}, S_0) + \sum_{j=1}^n w_j cov(\mathbf{w}S^{(n)}, S_j)$$

on a

$$S_i = S_0 + \sum_{j=1}^i (S_j - S_{j-1}), \quad i = 1, \dots, n, \quad (2.15)$$

donc

$$\sum_{j=1}^n w_j cov(\mathbf{w}S^{(n)}, S_j) = \sum_{j=1}^n w_j cov(\mathbf{w}S^{(n)}, S_0 + \sum_{k=1}^j (S_k - S_{k-1})) \quad (2.16)$$

$$= \sum_{j=1}^n w_j cov(\mathbf{w}S^{(n)}, S_0) + \quad (2.17)$$

$$\sum_{j=1}^n \sum_{k=1}^j w_j cov(\mathbf{w}S^{(n)}, S_k - S_{k-1}) \quad (2.18)$$

d'un autre coté, on a que

$$cov(\mathbf{w}S^{(n)}, S_k - S_{k-1}) = 0 \quad k = 1, \dots, n, \quad (2.19)$$

donc

$$\sum_{j=1}^n w_j \text{cov}(\mathbf{w}S^{(n)}, S_j) = \sum_{j=1}^n w_j \text{cov}(\mathbf{w}S^{(n)}, S_0) \quad (2.20)$$

$$= \text{cov}(\mathbf{w}S^{(n)}, S_0) \sum_{j=1}^n w_j \quad (2.21)$$

$$\begin{aligned} \text{cov}(\mathbf{w}S^{(n)}, S_0) &= \text{cov}\left(\sum_{i=0}^n w_i S_i, S_0\right) = w_0 \text{cov}(S_0, S_0) + \sum_{i=1}^n w_i \text{cov}(S_i, S_0) \\ &= w_0 \sigma_0^2 + \sum_{i=1}^n w_i \text{cov}\left(S_0 + \sum_{k=1}^i (S_k - S_{k-1}), S_0\right) \\ &= w_0 \sigma_0^2 + \sum_{i=1}^n w_i \text{cov}(S_0, S_0) + \sum_{i=1}^n \sum_{k=1}^i w_i \text{cov}(S_k - S_{k-1}, S_0) \\ &= \sigma_0^2 + \sum_{i=1}^n \sum_{k=1}^i w_i r_k. \end{aligned}$$

donc

$$\begin{aligned} \text{var}(\mathbf{w}S^{(n)}) &= w_0 \text{cov}(\mathbf{w}S^{(n)}, S_0) + \text{cov}(\mathbf{w}S^{(n)}, S_0) \sum_{j=1}^n w_j = \text{cov}(\mathbf{w}S^{(n)}, S_0) \sum_{j=0}^n w_j \\ &= \text{cov}(\mathbf{w}S^{(n)}, S_0) = \sigma_0^2 + \sum_{i=1}^n \sum_{j=1}^i w_i r_j \end{aligned}$$

De la proposition précédente on a :

$$r_n = -\sigma_I^2 w_n \quad (2.22)$$

$$r_j = \sigma_I^2 w_{j-1} + r_{j-1} = \sigma_I^2 w_{j-1} + \dots + \sigma_I^2 w_1 + r_1 \quad (2.23)$$

$$= \sigma_I^2 \sum_{k=1}^{j-1} w_k + r_1, \quad j = 2, \dots, n, \quad (2.24)$$

Pour $j = n$ on a

$$r_n = \sigma_I^2 \sum_{k=1}^{n-1} w_k + r_1$$

donc

$$r_1 = -\sigma_I^2 \sum_{k=1}^{n-1} w_k - \sigma_I^2 w_n = -\sigma_I^2 \sum_{k=1}^n w_k$$

et donc

$$r_j = -\sigma_I^2 \sum_{k=j}^n w_k, \quad j = 1, \dots, n, \quad (2.25)$$

finalement

$$\sigma_w^2 = \text{var}(\mathbf{w}S^{(n)}) = \sigma_0^2 - \sigma_I^2 \sum_{i=1}^n w_i \left\{ \sum_{j=1}^i \left(\sum_{k=j}^n w_k \right) \right\} \quad (2.26)$$

$$= \sigma_0^2 - \sum_{i=1}^n \frac{r_i^2}{\sigma_I^2}, \quad (2.27)$$

ce qui termine la preuve. □

Proposition 2.5.

$$\sigma_0^2 - \sum_{i=1}^n \frac{r_i^2}{\sigma_I^2} \geq 0$$

Proposition 2.6. *Le MLE des paramètres $\sigma_0^2, r_1, \dots, r_n$ et m_0 sont solutions du système*

$$\sigma_0^2 - \sum_{i=1}^n \frac{r_i^2}{\sigma_I^2} = 0,$$

et

$$m_0 = \sum_{i=0}^n w_i s_i - m_I \sum_{i=1}^n i w_i$$

Démonstration. Pour estimer les paramètres, il suffit de considérer la loi de

$$\sum_{i=0}^n w_i S_i$$

c'est une gaussienne de moyenne $\sum_{i=0}^n w_i (m_0 + i m_I) = m_0 + m_I \sum_{i=1}^n i w_i$ et de variance

$$\sigma_0^2 - \sum_{i=1}^n \frac{r_i^2}{\sigma_I^2}.$$

Le ML est égal à

$$m_0 + m_I \sum_{i=1}^n i w_i = \sum_{i=0}^n w_i s_i,$$

et

$$\sigma_0^2 - \sum_{i=1}^n \frac{r_i^2}{\sigma_I^2} = 0.$$

□

2.7 Algorithme d'Anderson

Proposition 2.7. *La matrice de covariance \mathbf{C} du vecteur aléatoire $(S_0, S_1 - S_0, \dots, S_n - S_{n-1})$ a la forme suivante :*

$$\sigma_0^2 \mathbf{e}_0 \mathbf{e}_0^\top + \sum_{i=1}^n r_i \{ \mathbf{e}_0 \mathbf{e}_i^\top + \mathbf{e}_i \mathbf{e}_0^\top \} + \sigma_I^2 \mathbf{diag}(0, 1, \dots, 1) \quad (2.28)$$

$$:= \sigma_0^2 \mathbf{G}_0 + \sum_{i=1}^n r_i \mathbf{G}_i + \sigma_I^2 \mathbf{diag}(0, 1, \dots, 1), \quad (2.29)$$

$$\mathbf{C} = \begin{pmatrix} \sigma_0^2 & r_1 & \dots & \dots & \dots & r_n \\ r_1 & \sigma_I^2 & 0 & \dots & \dots & 0 \\ \vdots & 0 & \sigma_I^2 & 0 & \dots & 0 \\ \vdots & \dots & \ddots & \ddots & \ddots & \vdots \\ r_n & 0 & \dots & \dots & \dots & \sigma_I^2 \end{pmatrix} \quad (2.30)$$

avec σ_0^2 qui dénote la variance de S_0 et r_i denote la covariance de $(S_0, S_i - S_{i-1})$. Ici \mathbf{e}_i denote le $i + 1$ -ème vecteur colonne canonique de \mathbb{R}^{n+1} .

La fonction -2 fois log vraisemblance des données $(s_0, s_1 - s_0, \dots, s_n - s_{n-1})$ a la forme

$$l(m_0, \sigma_0^2, r_1, \dots, r_n) = (n + 1) \ln(2\pi \det(\mathbf{C})) + (\mathbf{d} - m_0 \mathbf{e}_0)^\top \mathbf{C}^{-1} (\mathbf{d} - m_0 \mathbf{e}_0), \quad (2.31)$$

avec

$$\mathbf{d} = (s_0, s_1 - s_0 - m_I, \dots, s_n - s_{n-1} - m_I)^\top. \quad (2.32)$$

La -2 log-vraisemblance des données \mathbf{d} est égale à

$$\begin{aligned} \ell(\sigma_0^2, m_0, \mathbf{r}) &= (n + 1) \ln(2\pi) + \ln[\det(\mathbf{C})] \\ &\quad + \text{trace}\{(\mathbf{d} - m_0 \mathbf{e}_0)(\mathbf{d} - m_0 \mathbf{e}_0)^\top \mathbf{C}^{-1}\}, \end{aligned} \quad (2.33)$$

avec

$$\mathbf{C} = \sigma_0^2 \mathbf{G}_0 + \sum_{i=1}^n r_i \mathbf{G}_i + \sigma_I^2 \mathbf{diag}(0, 1, \dots, 1) =: \sum_{i=0}^n r_i \mathbf{G}_i + \sigma_I^2 \mathbf{diag}(0, 1, \dots, 1). \quad (2.34)$$

Proposition 2.8. *Les équations de vraisemblance*

$$\partial_{m_0}\ell = 0, \quad \partial_{r_i}\ell = 0, i = 0, \dots, n, \quad (2.35)$$

sont équivalentes, à

$$\begin{aligned} 1) m_0 &= \frac{\mathbf{e}_0^\top \mathbf{C}^{-1}(\mathbf{r})(\mathbf{d} - m_0 \mathbf{e}_0)}{\mathbf{e}_0^\top \mathbf{C}^{-1}(\mathbf{r})\mathbf{e}_0} \\ 2) \text{Tr}(\mathbf{C}^{-1}(\mathbf{r})\mathbf{G}_i) &= \text{Tr}\{(\mathbf{d} - m_0 \mathbf{e}_0)(\mathbf{d} - m_0 \mathbf{e}_0)^\top \mathbf{C}^{-1}(\mathbf{r})\mathbf{G}_i \mathbf{C}^{-1}(\mathbf{r})\}. \end{aligned}$$

Preuve 2. *La preuve de 2)*

On a

$$\partial \ell_{r_i} = 0$$

On a donc :

$$\partial(\ln(\det(C)) + \text{Tr}[(\mathbf{d} - m_0)(\mathbf{d} - m_0 \mathbf{e}_0)^\top \partial_{r_i}(\mathbf{C}^{-1})]) = 0$$

Or :

$$\partial(\ln(\det(C))) = \text{Tr}(\text{Com}(C)^\top \partial_{r_i} \mathbf{C}) / [\det(C)]^{-1}$$

ce qui donne :

$$\partial(\ln(\det(C))) = \text{Tr}(\text{Com}(C)^\top \mathbf{G}_i) [\det(C)]^{-1} = \text{Tr}(\mathbf{C}^{-1} \mathbf{G}_i)$$

car on a $\text{Com}(C)^\top * [\det(C)]^{-1} = \mathbf{C}^{-1}$ Et d'un autre côté : $\partial_{r_i}(\mathbf{C}^{-1}) = -\mathbf{C}^{-1}(\mathbf{r})\mathbf{G}_i \mathbf{C}^{-1}(\mathbf{r})$
d'où le résultat.

L'algorithme d'Anderson :

On note $r_{n+1} = \sigma_I^2$

$$\sum_{j=0}^{n+1} r_j(k+1) \text{Tr}\{\mathbf{C}^{-2}(\mathbf{r}(k))\mathbf{G}_j \mathbf{G}_i\} = \quad (2.36)$$

$$\text{Tr}\{(\mathbf{d} - m_0(k)\mathbf{e}_0)(\mathbf{d} - m_0(k)\mathbf{e}_0)^\top \mathbf{C}^{-1}(\mathbf{r}(k))\mathbf{G}_i \mathbf{C}^{-1}(\mathbf{r}(k))\}, \quad (2.37)$$

$$i = 0, \dots, n+1, \quad (2.38)$$

$$m_0(k+1) = \mathbf{w}(k+1)\mathbf{s}^{(n)} - m_I \mathbf{w}(k+1)(0, \dots, n)^\top, \quad (2.39)$$

sous la contrainte

$$\sigma_0^2 \sigma_I^2 > \sum_{i=1}^n r_i^2. \quad (2.40)$$

2.8 Cas particulier

2.8.1 Quelques propriétés

Dans le cas $r_2 = \dots = r_n = 0$ On estime la valeur de m_0 en calculant la valeur de m_0 qui maximise la vraisemblance du couple $(S_0, S_1 - S_0)$. La matrice de covariance de ce couple est notée Dans la suite pour les notations on note que $r_2 = \sigma_I^2$, $r_0 = \sigma_0^2$.

$$A = \sigma_0^2 \mathbf{G}_0 + r_1 \mathbf{G}_1 + \sigma_I^2 \mathbf{G}_I := r_0 \mathbf{G}_0 + r_1 \mathbf{G}_1 + r_2 \mathbf{G}_2.$$

$$Q(m_0) = (s_0 - m_0, s_1 - s_0 - m_I) A^{-1} (s_0 - m_0, s_1 - s_0 - m_I)^t \quad (2.41)$$

On a déjà estimé \hat{m}_I et

$$A = \begin{pmatrix} \sigma_0^2 & r_1 \\ r_1 & \sigma_I^2 \end{pmatrix} \quad (2.42)$$

on a donc

$$A^{-1} = \frac{1}{\sigma_0^2 \sigma_I^2 - r_1^2} \begin{pmatrix} \sigma_I^2 & -r_1 \\ -r_1 & \sigma_0^2 \end{pmatrix} \quad (2.43)$$

Proposition 2.9. *L'estimateur de m_0 est donné par :*

$$\hat{m}_0 = s_0 - \frac{r_1}{\sigma_I^2} (s_1 - s_0 - m_I) \quad (2.44)$$

Preuve 3. *On a :*

$$\begin{aligned} Q(m_0) &= (s_0 - m_0, s_1 - s_0 - m_I) \frac{1}{\sigma_0^2 \sigma_I^2 - r_1^2} \begin{pmatrix} \sigma_I^2 & -r_1 \\ -r_1 & \sigma_0^2 \end{pmatrix} (s_0 - m_0, s_1 - s_0 - m_I)^t \\ &= \frac{1}{\det A} (s_0 - m_0, s_1 - s_0 - m_I) \begin{pmatrix} \sigma_I^2 (s_0 - m_0) - r_1 (s_1 - s_0 - m_I) \\ -r_1 (s_0 - m_0) + \sigma_0^2 (s_1 - s_0 - m_I) \end{pmatrix} \\ &= \frac{1}{\det(A)} [\sigma_I^2 (s_0 - m_0)^2 - 2r_1 (s_1 - s_0 - m_I) (s_0 - m_0) + \sigma_0^2 (s_1 - s_0 - m_I)^2] \end{aligned}$$

Pour obtenir, le minimum de $Q(m_0)$, on va prendre le m_0 qui annule la dérivée : $\partial_{m_0} Q(m_0) = 0$. Ce qui donne :

$$\partial_{m_0} Q(m_0) = 0 \quad (2.45)$$

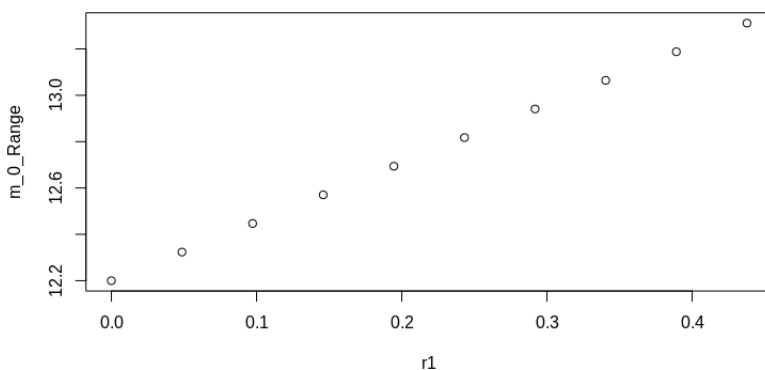
$$\iff \frac{1}{\det(A)} (-2\sigma_I^2 (s_0 - m_0) + 2r_1 (s_1 - s_0 - m_I)) = 0 \quad (2.46)$$

$$m_0 = s_0 - \frac{r_1}{\sigma_I^2} (s_1 - s_0 - m_I). \quad (2.47)$$

Remarque 5

On a par ailleurs par cauchy schwarz que : $r_1 = Cov(S_0, S_1 - S_0) = E(S_0(S_1 - S_0)) \leq E(S_0^2)E((S_1 - S_0)^2) \leq \sigma_0^2\sigma_I^2$. A partir de cela on peut faire le graphique de m_0 en fonction de r_1 pour par exemple la valeur de $\sigma_0 = 1$.

En appliquant cette estimateur \hat{m}_0 aux variables de $r_1 \in [0, \sigma_I^2]$ on obtient une droite. (ce qui est logique car m_0 dépend linéairement de r_1 .)



Le code pour l'obtention de ce graphe est dans l'annexe B.4.

2.8.2 Application de l'algorithme d'anderson

On applique à ce cas de figure l'algorithme d'anderson. L'algorithme est fournit en Annexe B.5 .L'objectif est d'obtenir une estimation de r_1, σ_0, m_0 . On fournit en entrée une intitialisation de ces quantités et l'algorithme nous donne les estimations en sortie. Les équations vus à la section précédente donnent : $r_2 = \sigma_I^2$

$$\begin{aligned}
& r_0(k+1)\text{Tr}\{A^{-2}(\mathbf{r}(k))\mathbf{G}_0^2\} + r_1(k+1)\text{Tr}\{A^{-2}(\mathbf{r}(k))\mathbf{G}_1\mathbf{G}_0\} + r_2\text{Tr}\{A^{-2}(\mathbf{r}(k))\mathbf{G}_2\mathbf{G}_0\} \\
& \quad = \text{Tr}\{(\mathbf{d} - m_0(k)\mathbf{e}_0)(\mathbf{d} - m_0(k)\mathbf{e}_0)^\top \mathbf{C}^{-1}(\mathbf{r}(k))\mathbf{G}_0\mathbf{C}^{-1}(\mathbf{r}(k))\}, \\
& r_0(k+1)\text{Tr}\{A^{-2}(\mathbf{r}(k))\mathbf{G}_0\mathbf{G}_1\} + r_1(k+1)\text{Tr}\{A^{-2}(\mathbf{r}(k))\mathbf{G}_1^2\} + r_2\text{Tr}\{A^{-2}(\mathbf{r}(k))\mathbf{G}_2\mathbf{G}_1\} \\
& \quad = \text{Tr}\{(\mathbf{d} - m_0(k)\mathbf{e}_0)(\mathbf{d} - m_0(k)\mathbf{e}_0)^\top A^{-1}(\mathbf{r}(k))\mathbf{G}_1\mathbf{C}^{-1}(\mathbf{r}(k))\}, \\
& \quad m_0(k+1) = \mathbf{w}(k+1)\mathbf{s}^{(n)} - m_I\mathbf{w}(k+1)(0, \dots, n)^\top,
\end{aligned}$$

sous la contrainte

$$\sigma_0^2(k+1)\sigma_I^2 > \sum_{i=1}^n r_i^2(k+1). \tag{2.48}$$

Dans le code on a mis le problème sous la forme :

$$Mx = b \quad (2.49)$$

Avec M une matrice 2, 2 avec $M[1, 1] = \text{Tr}\{A^{-2}(\mathbf{r}(k))\mathbf{G}_0^2\}$

$$M[1, 2] = \text{Tr}\{A^{-2}(\mathbf{r}(k))\mathbf{G}_1\mathbf{G}_0\}$$

$$M[2, 1] = \text{Tr}\{A^{-2}(\mathbf{r}(k))\mathbf{G}_0\mathbf{G}_1\}$$

$$M[2, 2] = \text{Tr}\{A^{-2}(\mathbf{r}(k))\mathbf{G}_1^2\}$$

Et

$$b1 = -r_2 \text{Tr}\{A^{-2}(\mathbf{r}(k))\mathbf{G}_2\mathbf{G}_0\} + \text{Tr}\{(\mathbf{d}-m_0(k)\mathbf{e}_0)(\mathbf{d}-m_0(k)\mathbf{e}_0)^\top \mathbf{C}^{-1}(\mathbf{r}(k))\mathbf{G}_0\mathbf{C}^{-1}(\mathbf{r}(k))\}$$

et

$$b2 = -r_2 \text{Tr}\{A^{-2}(\mathbf{r}(k))\mathbf{G}_2\mathbf{G}_1\} + \text{Tr}\{(\mathbf{d}-m_0(k)\mathbf{e}_0)(\mathbf{d}-m_0(k)\mathbf{e}_0)^\top A^{-1}(\mathbf{r}(k))\mathbf{G}_1\mathbf{C}^{-1}(\mathbf{r}(k))\}$$

.

On a fait tourner l'algorithme pour plusieurs initialisations de valeurs voici les résultats : pour $\sigma_0(0) = 1$ et $r_1(0) = 0.15$ on a :

$$\hat{\sigma}_0 = 0.4366629, \quad \hat{r}_1 = 0.06316923, \quad \hat{m}_0 = 11.84106$$

pour $\sigma_0(0) = 0.9$ et $r_1(0) = 0.2$ on a :

$$\hat{\sigma}_0 = 0.6335305, \quad \hat{r}_1 = 0.1537061, \quad \hat{m}_0 = 11.32661$$

pour $\sigma_0(0) = 0.7$ et $r_1(0) = 0.3$ on a :

$$\hat{\sigma}_0 = 0.7060538, \quad \hat{r}_1 = 0.3089209, \quad \hat{m}_0 = 10.44$$

Ci dessous se trouve le pseudo code :

Algorithm 1 Calculate σ_0, r_1, m_0

Require: $\sigma_0init, r_1init,$

```
 $\sigma_0 \leftarrow \sigma_0init;$ 
 $r_1 \leftarrow r_1init;$ 
 $m_I \leftarrow 0.0112069;$ 
 $m_0 \leftarrow s(115)[1] - r_1/\sigma_I^2 * (s(115)[2] - s(115)[1] - m_I);$ 
 $\sigma_0sol \leftarrow \sigma_0;$ 
 $r_1sol \leftarrow r_1;$ 
 $m_0sol \leftarrow m_0;$ 
 $k \leftarrow 0;$ 
 $continue \leftarrow TRUE;$ 
while  $(0 \leq r_1^2/\sigma_I^2) \& (\sigma_0^2 \geq r_1^2/\sigma_I^2) \& (continue)$  do
   $\sigma_0sol \leftarrow \sigma_0;$ 
   $r_1sol \leftarrow r_1;$ 
   $m_0sol \leftarrow m_0;$ 
   $M \leftarrow matrix(rep(0), nrow = 2, ncol = 2);$ 
   $b \leftarrow matrix(rep(0), nrow = 2, ncol = 1);$  calcul des matrices  $M_{ij}$ 
   $invA \leftarrow matrix(c(\sigma_I^2, -r_1, -r_1, sigma_0^2), ncol = 2, nrow = 2)$ 
   $invA \leftarrow 1/(sigma_0^2 * sigma_I^2 - r_1^2) * invA$ 
   $M[1, 1] \leftarrow tr(invA * invA * G0 * G0);$ 
   $M[1, 2] \leftarrow tr(invA * invA * G1 * G0); M[2, 1] \leftarrow tr(invA * invA * G0 * G1);$ 
   $M[2, 2] \leftarrow tr(invA * invA * G1 * G1);$  calcul des coefficients de  $b$ 
   $b[1] \leftarrow (-1) * \sigma_I^2 * tr(invA * invA * G2 * G0) + tr((d - m_0 * e0) * t(d - m_0 * e0) * invA * G0 * invA);$ 
   $b[2] \leftarrow (-1) * \sigma_I^2 * tr(invA * invA * G2 * G1) + tr((d - m_0 * e0) * t(d - m_0 * e0) * invA * G1 * invA);$ 
  calcul de  $sigma_0^2etr1$ 
   $solution \leftarrow -solve(M, b)$ 
  if  $solution[2] > 0$  then
     $r_1 \leftarrow solution[2]$ 
     $\sigma_0 \leftarrow sqrt(solution[1])$ 
     $m_0 \leftarrow s(115)[1] - solution[2]/\sigma_I^2 * (s(115)[1] - s(115)[2] - m_I)$ 
  else
     $continue = FALSE;$ 
  end if
   $k = k + 1$ 
end while
return  $list(\sigma_0sol, r_1sol, m_0sol);$ 
```

Partie 3

Arbre de régression

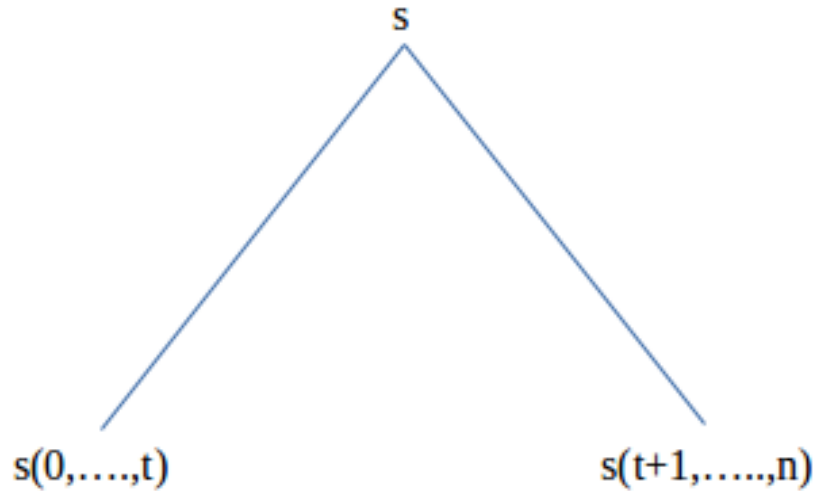
3.1 Théorie sur les arbres de régression

Dans l'étude de nos données on s'intéresse à la variable explicative(date) que l'on

note T définie par $\begin{pmatrix} t_0 \\ t_1 \\ \vdots \\ t_i \\ \vdots \\ t_{115} \end{pmatrix}$ et la variable expliquée (température) notée s et définie

par $\begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ s_i \\ \vdots \\ s_{115} \end{pmatrix}$.

Définition 3.1. Un arbre de régression est un outil d'aide à la décision représentant un ensemble de choix sous la forme graphique d'un arbre. Les différentes décisions possibles sont situées aux extrémités des branches (les « feuilles » de l'arbre), et sont atteints en fonction de décisions prises à chaque étape. L'arbre de décision est un outil utilisé dans des domaines variés tels que la sécurité, la fouille de données, la médecine, etc. Il a l'avantage d'être lisible et rapide à exécuter. Il s'agit de plus d'une représentation calculable automatiquement par des algorithmes d'apprentissage supervisé.



1.png

Dans ce contexte :

s c'est la moyenne empirique de l'échantillon.

s c'est la moyenne empirique des t premières températures.

$s(\mathbf{t+1}, \dots, \mathbf{n})$ c'est la moyenne empirique des températures de $t+1$ jusqu'à n .

Avoir un arbre très profond peut permettre de réduire au maximum les erreurs sur les données qui servent à le construire, mais peut par la suite ne plus être pertinent lors de l'ajout de nouvelles données. On utilise une techniques d'élagage pour réduire le nombre de noeuds.

Définition 3.2. Soit $\alpha > 0$, le paramètre de complexité. On pénalise l'erreur par un terme qui associe une importance aux nombres de noeuds :

$$R_\alpha(T) = R(T) + \alpha |T| \tag{3.1}$$

$|T|$ est le nombre de noeuds terminaux de T .

$T_\alpha = \operatorname{argmin}(R_\alpha(T) : T)$; $T_0 = T$ est l'arbre optimal le plus profond. $t_\infty = t(0)$ est la racine.

Le théorème suivant est un résultat qui permet de mettre en valeur le fait que pour des paramètres de complexité successifs, T_{α_2} est inclut dans T_{α_1} tel que $\alpha_1 < \alpha_2$. Inclut dans le sens où c'est un sous arbre.

Théorème 3.1. *Théorème de Breimann*

(1) T_α est un sous-arbre de T_0

(2) $\exists \alpha_0 = 0 < \alpha_1 < \alpha_2 < \dots < \alpha_{m-1} < \alpha_m = +\infty$

$T_\alpha = T_{\alpha_i} \forall \alpha \in [\alpha_i, \alpha_{i+1} [$

3.1.1 Algorithme de prédiction par un modèle à 2 feuilles

L'algorithme suivant est la réalisation d'un arbre à deux branches. Le paramètre qui scinde les données en deux sous groupes est celui de l'année. Pour un paramètre, ceux dont l'année est $\leq \theta$ sont dans le groupe 1, les autres dans le groupe 2. Chaque sous groupe aura une prévision correspondante à la moyenne de température de ce sous groupe. Le but est de résoudre le problème suivant :

$$\operatorname{argmin}_{t \in \{0, \dots, n\}} \left\{ \sum_{i=0}^t \left(s(i) - \frac{\sum_{j=0}^t s_j}{t+1} \right)^2 + \sum_{i=t+1}^n \left(s(i) - \frac{\sum_{j=t+1}^n s_j}{n-t} \right)^2 \right\} \quad (3.2)$$

L'algorithme consiste à créer un vecteur Y de taille 115 qui prenne en $Y(\theta)$ pour chaque θ de 1 à 115.

$$\sum_{i=1}^{\theta} \left(s(i) - \sum_{j=1}^{\theta} \frac{s(j)}{\theta} \right)^2 + \sum_{i=\theta+1}^{116} \left(s(i) - \sum_{j=\theta+1}^{116} \frac{s(j)}{116-\theta} \right)^2 \quad (3.3)$$

Sur la figure suivante, on peut observer Y . L'application numérique a calculé comme valeur seuil 1988 ou $i = 89$.

Le code pour obtenir le seuil pour l'arbre à deux branches est dans l'annexe C.1. L'utilisation de différentes bibliothèques met en évidence des arbres différents. L'arbre obtenue avec "rpart" est affiché en figure suivante. Il a 10 noeuds. d'autre part l'arbre créé avec "treefit" est un sous arbre de l'arbre créé avec rpart. Le code est dans l'annexe C.3.

Arbre à 2 Branches

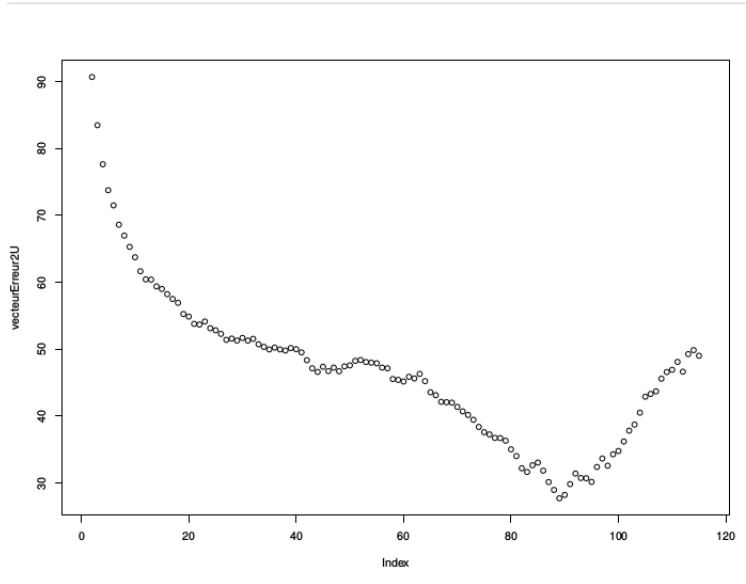
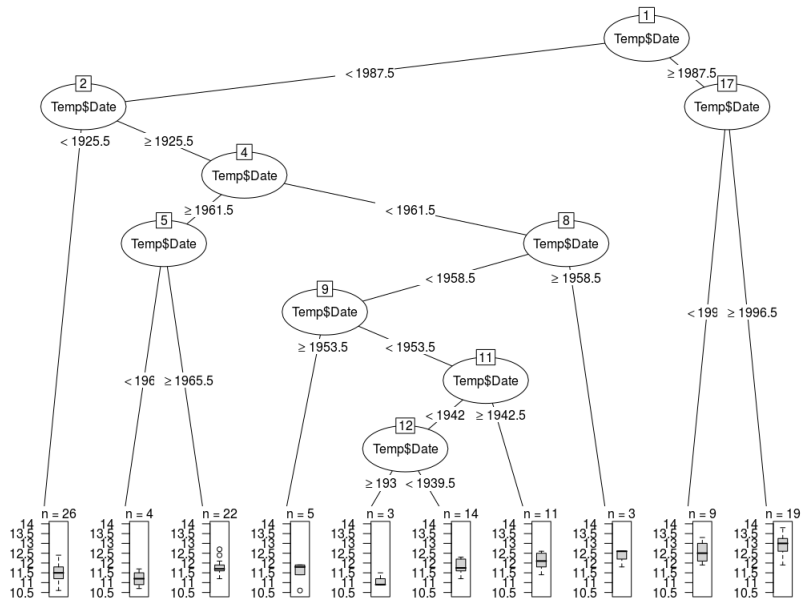
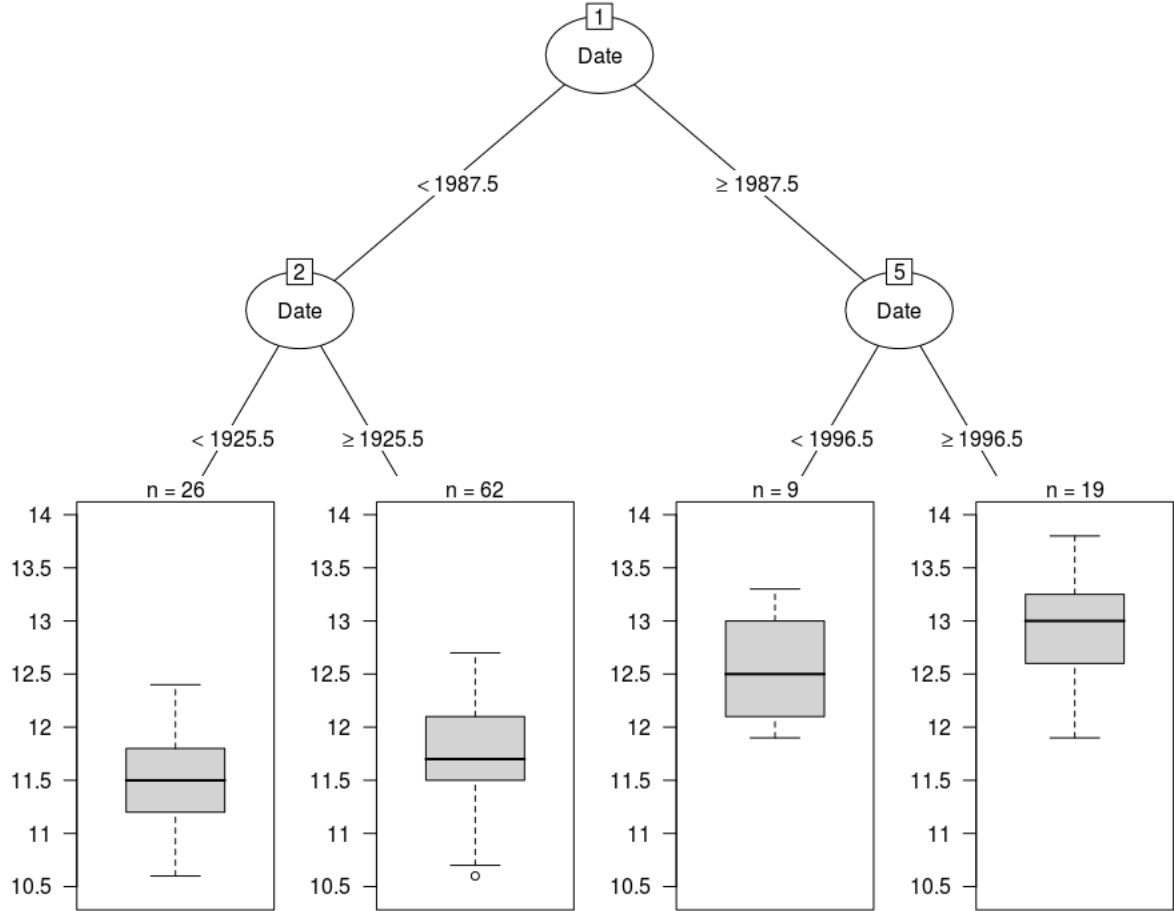


FIGURE 3.1 – arbre à 2 branches et erreur



Les noeuds 4,5,9,8,11, et 12 ont été enlevé dans le premier arbre ce qui donne le deuxième arbre suivant.



On peut afficher les paramètres de complexité pour voir quel est la partition la plus avantageuse :

	CP	nsplit	rel error	xerror	xstd
1	0.510599	0	1.00000	1.01348	0.127826
2	0.021962	1	0.48940	0.54076	0.066489
3	0.018952	2	0.46744	0.58174	0.065062
4	0.010000	3	0.44849	0.60613	0.071892

On observe que pour la minimisation de xerror est atteinte en $\alpha = 0.02156$ avec une seule division des données en deux groupes. Si on choisit de minimiser xstd on obtient $\alpha = 0.0118952$ et une division en trois groupes.

Le xerror minimal est 0.54076 qui correspond à un split et l'estimateur de \hat{s}_{n+1} est $\sum_{i=89}^n \frac{s_i}{n-89} = 12.8321$. Ce qui est logique car le groupe que l'on a calculé lors de l'algorithme de l'arbre à deux branches est conservé. Et de plus cette estimation tombe

dans l'intervalle de confiance calculé au chapitre sur la marche aléatoire. Le code pour l'estimation \hat{s}_{n+1} par l'arbre à deux branches est dans l'annexe C.2.

conclusion

Au cours de ce TER on a pu établir sur une base solide certaines propriétés concernant la variable aléatoire de la température annuelle moyenne : le premier modèle a mis en valeur une relation entre les 6 dernières années ce qui peut être par la suite approfondi par un modèle autoregressif d'ordre 6. Le modèle de marche aléatoire avec accroissement gaussien a permis de calculer un intervalle de confiance quant à la prédiction pour 2016 ainsi qu'une tendance linéaire. La température annuelle augmente en moyenne. D'autre part une analyse de la covariance de $S_0, S_1 - S_0, \dots, S_n - S_{n-1}$ a permis d'affiner les méthodes pour comprendre la variable S_0 ainsi que son espérance et que sa variance. D'autre part le modèle de l'arbre de régression est optimal pour une valeur du paramètre de complexité qui le scinde en deux feuilles, avec un groupe de 1900 à 1988 et un autre de 1989 à 2015.

Annexe A

Le code

A.1 Graphe des Températures moyennes en fonction de la date

```
plot(Temp$Date, Temp$Tmoy)
```

A.2 Estimation modèle 1

```
## fonction qui retourne le vecteur des TempMoy de 1900 à 1900+l  
s <- function(l){  
  ss = matrix(nrow =(l+1), ncol =1);  
  ss = Temp$Tmoy[1:(l+1)];  
  s= as.matrix(ss)  
  return(s);  
}
```

```
## fonction qui retourne le vecteur de taille l+1 avec des 0 de 1 à l-k+1 et 1/k pour  
pfunction <- function(l,k){  
  p = matrix(nrow=(l+1),ncol=1);  
  for (i in 1:(l+1))  
  {  
    if(i<=l-k+1){  
  
      p[i,1]=0;
```

```

    }
    if(i>l-k+1){

        p[i,1]=1/k;
    }
}
pfunction = p;
}

## E(k) correspond à la matrice de dim(l-1,1) : avec pour coord: e(l+1,min(k,l))
## il n'y a pas de prédiction pour  $\hat{s}_1 = s_0$ 
##vérifié sinn pour tt les autres  $\hat{s}_{l+1}$ 
E <- fonction(k){
    e =matrix(nrow = n-2, ncol=1 );
    for (l in 0:(n-2)){
        e[l] = abs(s(l+1)[l+2] - t(pfunction(l,min(k,l)))%*%s(l))
    }

    erreur = max(e)

    return(erreur)
}

vecteurE <- matrix(nrow = n-1, ncol =1);
for (u in 1:(n-1)){
    vecteurE[u,]<-E(u)
}
## on a pour valeur optimale usol =à et le koptimal = usol-1 car on a commencé en 0,
k_opti <- which.min(vecteurE)

##ici l'abscisse affiche u qui correspond à
plot(vecteurE, main = "Graphe de E(k) ",ylim = c(0,2.5))

```



```

print(plot(vecteurE))
##zoom sur la valeur k =6
plot(vecteurE, main = "Graphe de E(k) ",xlim = c(0,9))

```

A.3 Estimation de s_{116}

```

estimation_2016 =0;
for(i in 1:6)
{
  estimation_2016 = estimation_2016 + Temp$Tmoy[116-i]/6
}
print(estimation_2016)

```

A.4 test de gaussiannité des résidus

```

##qqplot des résidus
##fonction qui génère les résidus pour le choix de k
residus <- fonction(k){
  e=matrix(nrow = n-2, ncol=1 );
  for (l in 0:(n-2)){
    e[l] = s(l+1)[l+2] - t(pfunction(l,min(k,l)))%*%s(l)
  }

  return(e)
}

res<-residus(6)
plot(res)
library(car)
qqPlot(res,main="Qqplot des résidus")
mean(res)
var_res <-sd(res)^2
print(var_res)

```

Annexe B

Modèle de la Marche aléatoire

B.1 Test de gaussianité des incréments

```
ecartTemp = matrix(nrow=n-1);
ecartTemp = Temp[2:n,]-Temp[1:(n-1),];
plot(ecartTemp$Tmoy)
library(car)
qqPlot(ecartTemp$Tmoy)
abline(0,1)
##le résultat est gaussien

## variance et moyenne des Xi
summary(ecartTemp$Tmoy)
sd(ecartTemp$Tmoy)
##mean 0.0113 et écart types 0.6702664
```

B.2 estimation des paramètres m_i et σ_I^2

```
Moy_emp = (s(116)[116]-s(116)[1])/n

var <- fonction(n,moy_emp)
{
  var <-0
```

```

for (i in 1:115)
{
    var<- var + (s(116)[i+1] -s(116)[i]- moy_emp)^2/n
}

var
}

var_I = var(n,Moy_emp)

```

B.3 Intervalle de Confiance de s_{n+1}

```

##première borne et deuxième borne pour n =116
borne1 <- s(n)[n]+Moy_emp -1.96*sqrt(var_I)
borne2 <- s(n)[n] + Moy_emp + 1.96*sqrt(var_I)
Temp$Date[n]

```

B.4 Estimation de Tendence

```

## pour la valeur de  $m_I$  on utilise  $\hat{m}_I$  , l'estimateur de la moyenne pour 1
##pour la valeur de  $\sigma_I^2$  on utilise  $\hat{m}_I^2$ , l'estimateur pour le max de vr
##range de r1 pour  $\sigma_0 = 1$  et  $r1 \leq \sigma_0^2 * \sigma_I^2$ .

```

```

##on récupère dans var_I.R les estimateurs de  $m_i$  et  $\sigma_I^2$ 

```

```

r1<- seq(0,var_I,length.out = 10)
s0<-s(2)[1]
s1<-s(2)[2]
m_0_Range <- m0(s0, s1,r1,var_I,Moy_emp)
plot(m_0_Range,r1)

```

B.5 Algorithme d'Anderson

```
##Algorithme d'Anderson
##-----
setwd("/home/axel/Documents/Information/Matiere-Intellectuelle/Mathématiques/cours MA
Temp <-read.csv2("/home/axel/Documents/Information/Matiere-Intellectuelle/Mathématiqu
                dec = ",", fill = TRUE, comment.char = "")
library(psych)

##fonction s

s <- function(l){
  ss = matrix(nrow =(l+1), ncol =1);
  ss = Temp$Tmoy[1:(l+1)];
  s= as.matrix(ss)
  return(s);
}

##valeurs utilisées dans l'algorithme d'anderson
m_I<- 0.0112069;
sigma_I <- 0.437738;

##matrices utilisées dans AAD:
G0<-matrix(c(1,0,0,0), nrow=2,ncol=2)
G1<- matrix(c(0,1,1,0),nrow=2,ncol=2)
G2 <- matrix(c(0,0,0,1),nrow=2,ncol=2)

d<- matrix(nrow= 2,ncol=1)
d[1]<-s(115)[1]
d[2] <- s(115)[2]-s(115)[1]-m_I

e0<-matrix(c(1,0),nrow=2,ncol=1)

## algorithme AAD
```

```

AAD<-function(sigma_I, sigma0_init,r1_init){
  ##initialisation
  sigma_0 <-sigma0_init;
  r1 <- r1_init;
  m_I<- 0.0112069;
  m_0 <- s(115)[1]-r1/sigma_I^2*(s(115)[2]-s(115)[1]-m_I);

  sigma_0sol <- sigma_0;
  r1_sol <- r1;
  m_0_sol <- m_0;
  ##itération sur k
  k=0;
  continue = TRUE;

  while((0<=r1^2/sigma_I^2)&&(sigma_0^2>=r1^2/sigma_I^2)&&(continue))
  {

    ##sauvegarde des dernières données qui vérifie la contrainte
    sigma_0sol <- sigma_0;
    r1_sol <- r1;
    m_0_sol <- m_0;
    M <- matrix(rep(0),nrow=2,ncol=2);
    b<-matrix(rep(0),nrow=2,ncol=1);
    ##calcul des matrices Mij
    invA <-matrix(c(sigma_I^2,-r1,-r1,sigma_0^2),ncol=2,nrow=2)
    invA <- 1/(sigma_0^2*sigma_I^2-r1^2)*invA
    M[1,1]<- tr(invA%%invA%%G0%%G0);
    M[1,2]<-tr(invA%%invA%%G1%%G0);
    M[2,1]<-tr(invA%%invA%%G0%%G1);
    M[2,2]<-tr(invA%%invA%%G1%%G1);
    ##calcul des coefficients de b
    b[1]<- (-1)*sigma_I^2*tr(invA%%invA%%G2%%G0)+tr((d-m_0*e0)%*%t(d-m_0*e0)%*%inv
    b[2]<- (-1)*sigma_I^2*tr(invA%%invA%%G2%%G1)+tr((d-m_0*e0)%*%t(d-m_0*e0)%*%inv

    ##calcul de sigma_0^2 et r1

    solution<-solve(M,b)
    print(solution)
  }
}

```

```

##print(solution)

if(solution[2]>0){
  r1<-solution[2]
  sigma_0 <- sqrt(solution[1])
  m_0<-s(115)[1]- solution[2]/sigma_I^2*(s(115)[1]-s(115)[2]-m_I)
}
else{
  continue = FALSE;
}
k=k+1
}

list(sigma_0sol, r1_sol, m_0_sol);
}

##initialisation de sigma_0, r1, m_0
array_sigma_0 =c(1, 0.9 , 0.7)
array_r1 =c(0.15, 0.20, 0.3)

##application algo

T1 <- AAD(sigma_I,array_sigma_0[1],array_r1[1])
T2 <- AAD(sigma_I,array_sigma_0[2],array_r1[2])
T3 <- AAD(sigma_I,array_sigma_0[3],array_r1[3])
print(T1)
print(T2)
print(T3)

```

Annexe C

Arbre de Régression

C.1 Code arbre à deux feuilles

```
## arbre à deux branche avec coefficient optimal ##
erreur_arbre2feuilles <- fonction(t,nmax,nmin){

  S1=0;
  S2=0;
  moyenneBranche_gauche = 0;
  moyenneBranche_droite = 0;

  ## moyenne sur la branche gauche
  for (k in nmin:(t-1))
  {

    moyenneBranche_gauche = moyenneBranche_gauche +s(nmax-1)[k];

  }
  moyenneBranche_gauche = moyenneBranche_gauche/t

  ##moyenne sur la branche droite
  for (k in t:(nmax))
  {
    moyenneBranche_droite = moyenneBranche_droite +s(nmax-1)[k];
  }
  moyenneBranche_droite = moyenneBranche_droite/(nmax-t+1)
```

```

## somme des erreurs au carré sur la branche 1
for (i in nmin:(t-1))
{
  S1 = S1 + (s(nmax-1)[i]-moyenneBranche_gauche)**2
}

##somme des erreurs aux carré sur la branche 2
for (i in t:(nmax))
{
  S2 = S2 + (s(nmax-1)[i]-moyenneBranche_droite)**2
}
erreurTotale = S1 +S2;
return(erreurTotale);
}

nmax= 116;
nmin =1;
vecteurErreur = matrix(0,nrow = nmax, ncol=1);

for(i in 1:116)
{ print(i)
  vecteurErreur[i,1]<-erreur_arbre2feuilles(i,nmax,nmin)[1];
}
print(vecteurErreur)

plot(vecteurErreur)
t_opti <- which.min(vecteurErreur)

```

C.2 estimation de s_{n+1} avec le modèle de l'arbre à deux feuilles

```

moyenneBranche_droite_opti =0
for (k in t_opti:(nmax))

```



```
{
  moyenneBranche_droite_opti = moyenneBranche_droite_opti +s(nmax-1)[k];
}
```

```
moyenneBranche_droite_opti = moyenneBranche_droite_opti/(nmax-t_opti+1)
##la valeur de moyenneBranche_droite_opti = 12.8321
```

C.3 Code de l'arbre de régression

```
Temp$Tmoy
Temp_rpart<-rpart(Tmoy~Date,data=Temp,control=rpart.control(minisplit=10))
T
Temp_rpart
library("partykit")
plot(as.party(Temp_rpart),tp_args = list(id = FALSE))

print(Temp_rpart$cptable)
opt<-which.min(Temp_rpart$cptable[,"xerror"])
cp<-Temp_rpart$cptable[opt,"CP"]# c' est le calcul du parametre de complicité

Temp_rpart_prune<-prune(Temp_rpart,cp=cp)
Temp_rpart_prune

plot(as.party(Temp_rpart_prune),tp_args=list(id=FALSE))
```

Bibliographie

- [1] pdf Renforcement statistique, Séries Chronologiques, Agnès Lagnoux
- [2] 15mars.pdf, Azzouz Dermoune 1 , Khalifa Es-Sebaiy 2 , Mohammed Es.Sebaiy 3 and Jabrane Moustaaïd 4 Lille University and Cadi Ayyad University
- [3] Azzouz Dermoune 1 , Khalifa Es-Sebaiy 2 , Mohammed Es.Sebaiy 3 and Jabrane Moustaaïd 4 Lille University and Cadi Ayyad University