

GUILLOT Juliette
MARINGER Léa
ING2 GMI2



ETUDE DE LA SÉRIE DES TEMPÉRATURES MOYENNES MENSUELLES À SAN DIEGO (CA, ETATS-UNIS)

Rapport du mini-projet dans le cadre du module de Séries Temporelles.

*A l'attention de :
M. Laib NAAMANE*

31 mars 2023

Table des matières

Introduction	2
1 Description et caractéristiques	3
1.1 Série et tendance	3
1.2 Saisonnalité	4
1.3 Type de modèle	6
2 Désaisonnalisation de la série	9
2.1 Méthode de la moyenne mobile	9
2.2 Régression sur le temps avec des variables indicatrices	11
3 Prévisions par lissage exponentiel	14
3.1 Lissage exponentiel simple	14
3.2 Méthode de Holt Winters	15
3.3 Comparaison des deux méthodes	17
Annexes : code R	19

Introduction

Une série temporelle est une suite d'observations portant sur un même objet et recueillies à des instants équidistants. Mathématiquement, elle peut être vue comme une réalisation d'une famille de variables aléatoires ou comme un processus stochastique $(X_t$ avec $t \in \mathbb{Z}$). Les séries temporelles sont utilisées pour analyser des phénomènes qui évoluent au cours du temps comme les ventes d'une entreprise, la croissance de la population ou encore le cours de la bourse. Elles permettent alors de modéliser les tendances, les saisons, les cycles et les fluctuations aléatoires, ce qui peut aider à prévoir les valeurs futures et à comprendre les relations entre les différentes variables. Enfin, elles sont largement utilisées dans de nombreux domaines comme la finance, l'économie, la météorologie, l'ingénierie ou la santé publique.

Dans ce rapport, nous nous proposons d'analyser la série temporelle des températures moyennes mensuelles à San Diego (CA, Etats-Unis). En effet, les températures varient selon les saisons, et nous nous sommes également questionnées sur la croissance des températures due au réchauffement climatique. Après avoir décrit et analysé les caractéristiques de la série, nous lui appliquerons deux méthodes de désaisonnalisation. Enfin, nous effectuerons des prévisions grâce à deux méthodes de lissage exponentiel.

1 Description et caractéristiques

Vous trouverez les données dans le fichier *donnees.csv* ou directement sur le site [National Centers for Environmental Information](#) (lien cliquable). Les données représentent les températures moyennes mensuelles à San Diego de janvier 2000 à février 2023. Nous disposons donc de 278 observations mensuelles et nous avons décidé de convertir les données (initialement en degrés Fahrenheit) en degrés Celsius afin de faciliter l'interprétation des résultats.

1.1 Série et tendance

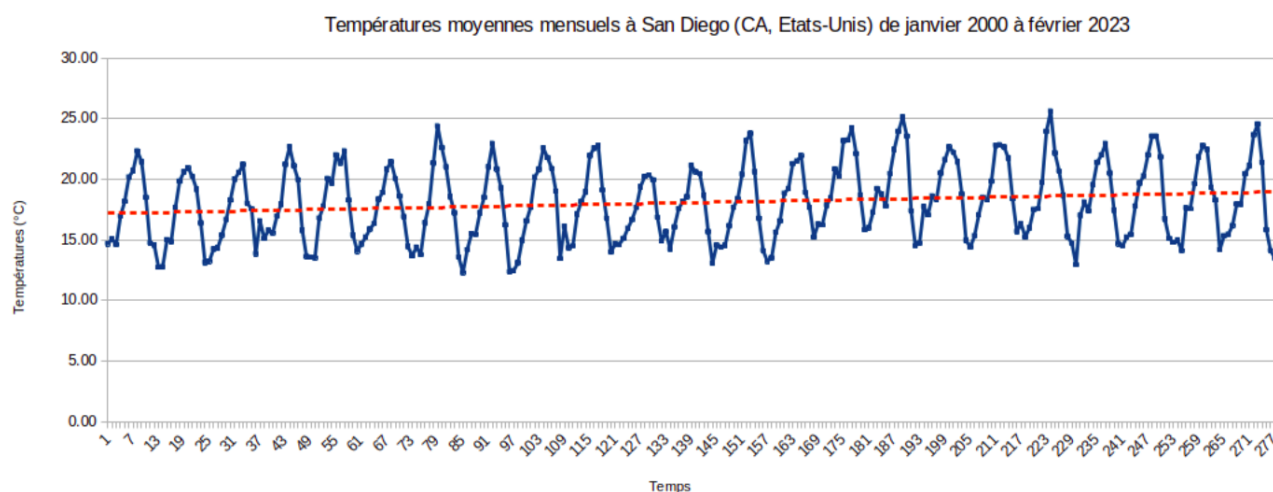


FIGURE 1 – Graphique de la série et de sa tendance en fonction du temps.

A première vue, nous remarquons qu'il y a une saisonnalité dans la série (voir 1.2). Nous notons également une légère tendance croissance. Pour confirmer cette hypothèse, nous avons tracé la tendance donnée par la fonction $TENDANCE()$ d'Excel. De plus, une régression linéaire afin estimer la température en fonction du temps donne exactement la même droite de tendance, dont l'équation est la suivante $m_t = a + b \cdot t$ avec $a = 17.168$ et $b = 0.006$, où X_t est la valeur de la série à l'instant t .

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	17.167564	0.371026	46.271	< 2e-16 ***
t	0.006426	0.002305	2.787	0.00569 **

FIGURE 2 – Coefficients donnés par la régression linéaire de la série sur le temps.

1.2 Saisonnalité

Le nombre important de données ne nous permettait pas de définir la période de la série à partir de la figure 1. Cependant, en zoomant sur le graphique et en n'affichant que quelques années, nous pouvons constater que la période de la série est de 12 mensualités, soit 1 année.

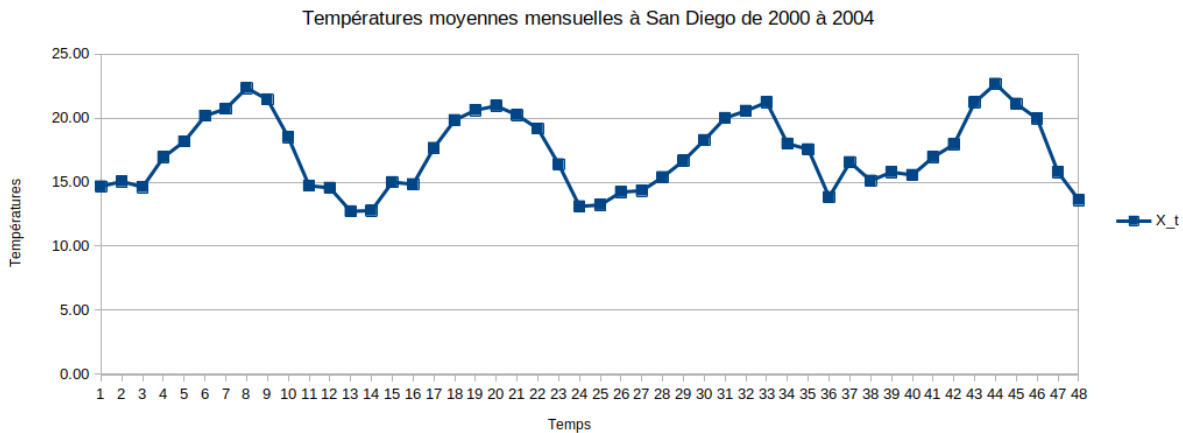


FIGURE 3 – Zoom sur le graphique de la série : températures moyennes mensuelles de 2000 à 2004.

Plusieurs analyses plus profondes nous permettent de confirmer cela. En effet, dans un premier temps, nous pouvons calculer la fonction d'autocorrélation grâce à la fonction $acf()$ de R. Le graphique donné montre bien une saisonnalité de période 12. De plus, la décroissance de la fonction confirme encore l'existence d'une tendance.

Aussi, nous avons dessiné les nuages de points des couples (X_t, X_{t+h}) pour $h = 1$, $h = 3$, $h = 6$ et $h = 12$ afin d'étudier la corrélation entre les observations aux instants t et aux instants $t + h$. Nous constatons qu'il existe une corrélation linéaire entre les couples (X_t, X_{t+12}) pour les différentes valeurs de t . Cela appuie encore notre affirmation précédente sur la saisonnalité annuelle.

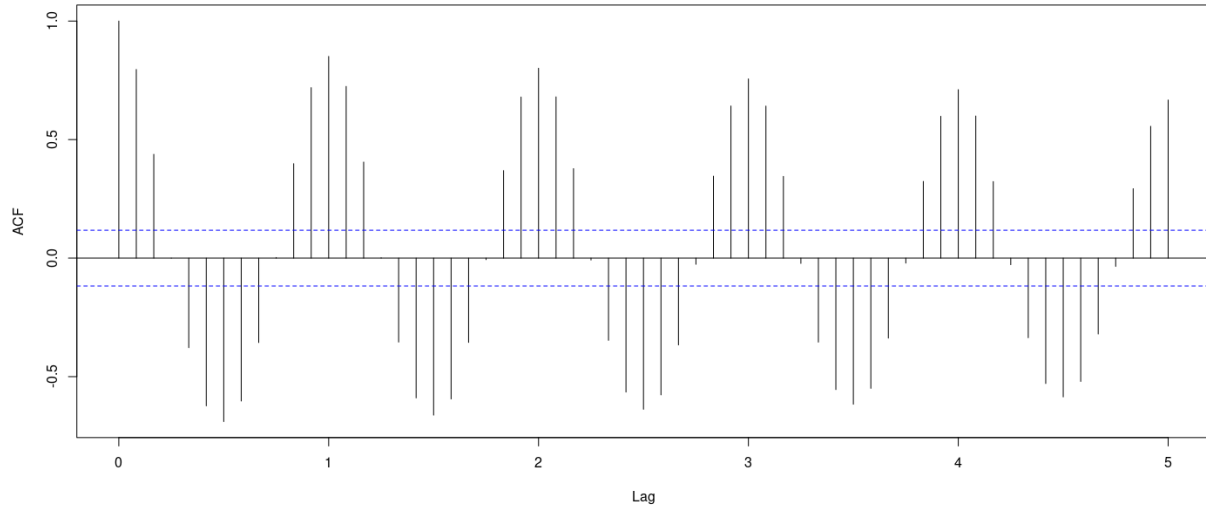
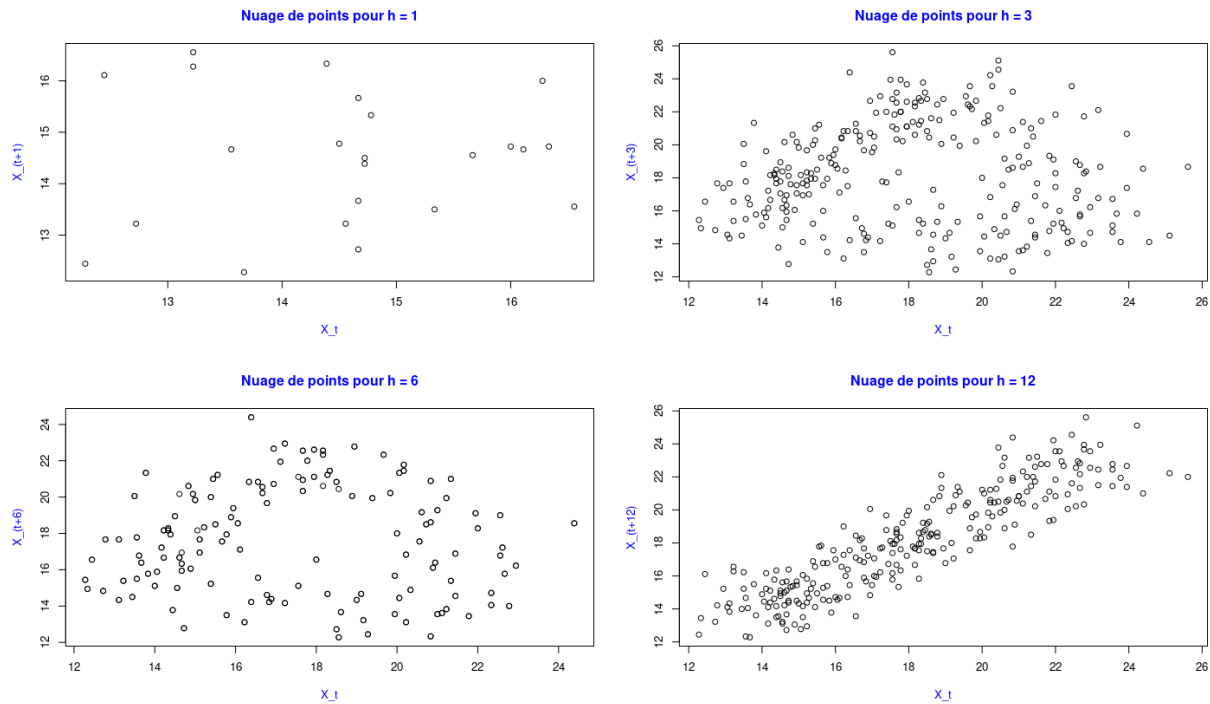


FIGURE 4 – Fonction d'autocorrélation des valeurs de la série.

FIGURE 5 – Nuages de points des couples (X_t, X_{t+h}) pour différentes valeurs de h .

Nous avons alors superposé les valeurs de la série pour chaque année afin de les comparer. Nous remarquons que pour chaque année les températures moyennes mensuelles forment

plus ou moins la même courbe. De plus, la température moyenne est entre 12 et 16 degrés Celsius en début d'année, croît jusqu'au mois de septembre pour atteindre une valeur entre 20 et 25 degrés Celsius, avant de décroître jusqu'au mois de janvier suivant.

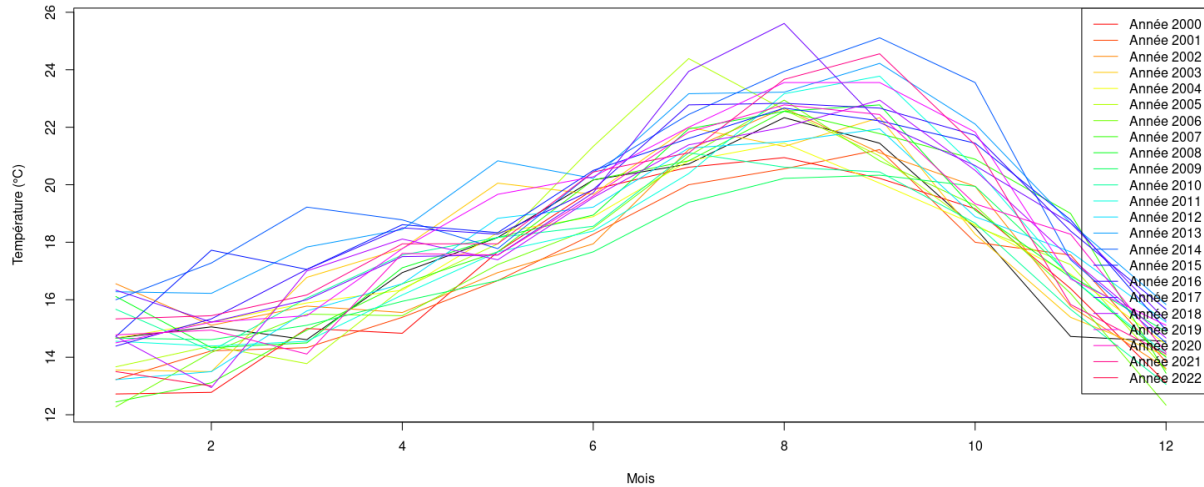


FIGURE 6 – Graphique de la série par année.

1.3 Type de modèle

Deux possibilités se présentent à nous pour modéliser la série. En effet nous pouvons utiliser :

- soit un modèle additif : $X_t = m_t + S_t + \epsilon_t$,
- soit un modèle multiplicatif : $X_t = m_t \cdot S_t \cdot \epsilon_t$,

où X_t est la température moyenne au mois t , m_t représente la tendance, S_t les coefficients saisonniers et ϵ_t les variations aléatoires. Notons qu'il existe d'autres types de modèles, mais nous ne nous attarderons pas sur ces derniers ici.

Le calcul des moyennes et des écart-types cumulés mensuels montre que la série varie de façon constante autour de la moyenne. De plus, nous constatons que la droite reliant les maximums de la série et celle reliant les minimums de la série sont quasiment parallèles, ce qui suggère qu'un modèle additif est plus adapté à la modélisation de la série qu'un modèle multiplicatif.

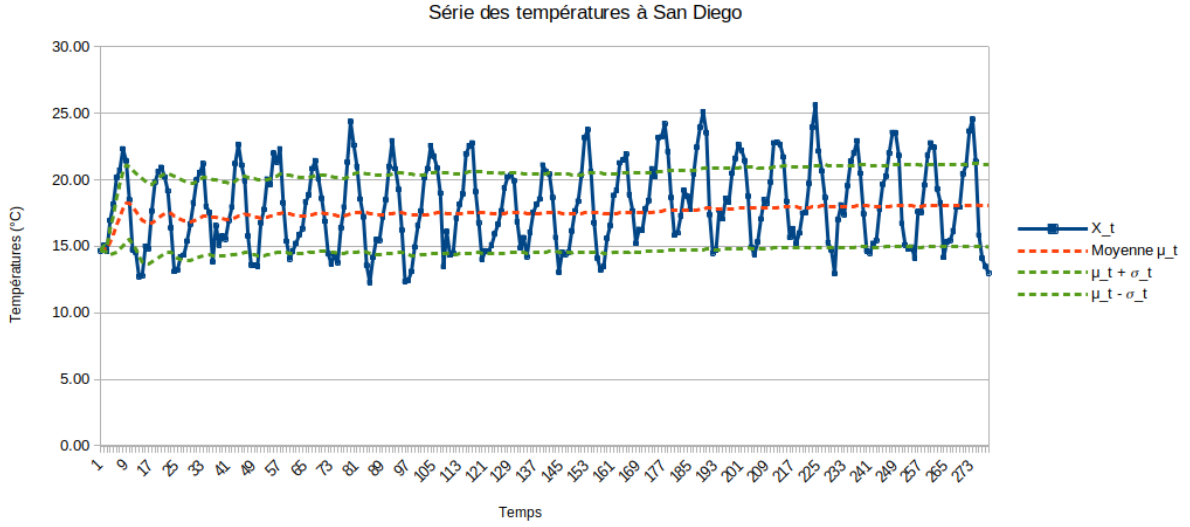


FIGURE 7 – Série, moyenne et variations autour de la moyenne.

Nous pouvons également étudier la corrélation entre les moyennes et les écart-types annuels. En effet, si ces derniers sont corrélés, cela indique qu'il est préférable de privilégier un modèle multiplicatif. Dans le cas contraire, nous utilisons un modèle additif.

Nous constatons que la moyenne et l'écart-type par année sont peu corrélés. Le calcul de leur coefficient de corrélation effectué sur Excel nous donne la corrélation suivante :

$$r = \frac{Cov(\mu_{an}, \sigma_{an})}{\sigma(\mu_{an}) \cdot \sigma(\sigma_{an})} = 0.42$$

où μ_{an} est la série des moyennes annuelles et σ_{an} la série des écart-types annuels. Ce coefficient reste relativement élevé mais pas suffisamment pour pouvoir envisager un modèle multiplicatif. Nous choisissons alors de modéliser la série comme suit :

$$X_t = m_t + S_t + \epsilon_t$$

Afin de pouvoir effectuer des prédictions dans les parties suivantes et les comparer aux valeurs réelles, nous séparons notre série en deux séries :

- une série d'entraînement (*train*) prenant les données de janvier 2000 à décembre 2017 (environ 80% de la série initiale),
- une série de test (*test*) allant de janvier 2018 à février 2023 (environ 20% de la série initiale).

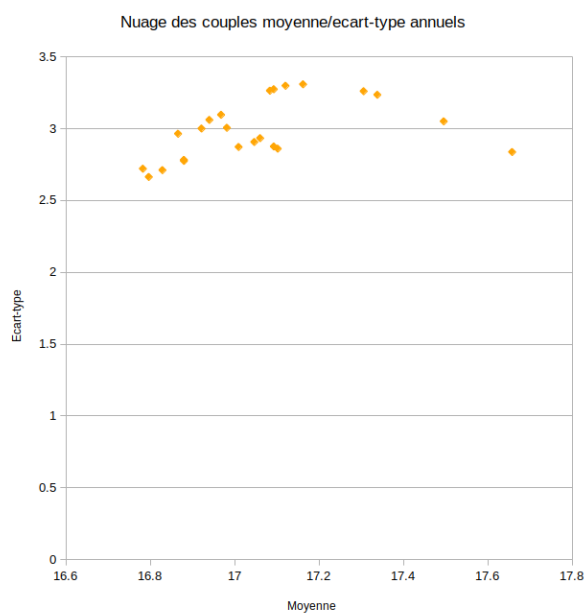


FIGURE 8 – Nuage de points des couples moyenne/écart-type par année.

2 Désaisonnalisation de la série

Nous nous intéressons maintenant à la désaisonnalisation de la série temporelle. En effet, la désaisonnalisation permet de séparer la tendance des variations saisonnières et ainsi de mieux comprendre les fluctuations de la série, mais elle permet surtout d'améliorer la précision des prédictions. Nous utiliserons ici deux méthodes.

2.1 Méthode de la moyenne mobile

Le principe du lissage par moyenne mobile appliqué à un modèle additif est le suivant : appliquer un filtre à notre série permettant de conserver la tendance, d'annuler la saisonnalité et de réduire la variance de la partie aléatoire.

Ici, nous choisissons une moyenne mobile symétrique d'ordre $p = 12$ puisque notre série contient une saisonnalité annuel (donc de 12 mois). La formule de la moyenne mobile est donc :

$$M_{12}(X_t) = \frac{1}{12} \cdot \left(\frac{1}{2} \cdot X_{t-6} + \dots + X_{t-1} + X_t + X_{t+1} + \dots + \frac{1}{2} \cdot X_{t+6} \right)$$

Cette transformation permet d'éliminer les saisonnalités mensuelles et conserve les tendances linéaires. Après avoir calculer la moyenne mobile de chaque valeurs de l'ensemble d'apprentissage (à l'exception des 6 premières et des 6 dernières que l'on ne peut pas calculer), nous soustrayons cette moyenne aux différents X_t .

On calcule alors les 12 coefficients saisonniers provisoires c_t . Pour le coefficient saisonnier de chaque mois, il s'agit de la moyenne de ses $C_t = X_t - M_{12}(X_t)$. Soit c_i le coefficient saisonnier du i -ème mois ($i \in \llbracket 1; 12 \rrbracket$). Alors :

$$c_i = E(\{C_k, k = i + 12 \cdot m \text{ avec } m \in \llbracket 0; 16 \rrbracket\})$$

Nous centrons ensuite ces coefficients de moyenne $\bar{c} = -0.0087$ puisque la somme des coefficients saisonniers doit être nulle. On a donc $c'_i = c_i - \bar{c}$.

	c_i	c'_i
Janvier	-3.537	-3.528
Février	-3.226	-3.218
Mars	-2.119	-2.111
Avril	-1.134	-1.125
Mai	0.101	0.110
Juin	1.334	1.343
Juillet	3.441	3.450
Août	4.167	4.175
Septembre	3.899	3.907
Octobre	1.814	1.822
Novembre	-1.004	-0.995
Décembre	-3.840	-3.831

Nous utilisons également la fonction *decompose()* de R qui nous donne des coefficients saisonniers quasiment identiques, à 10^{-3} près.

Nous retirons ensuite la saisonnalité à la série en retirant à chaque valeurs le coefficient saisonnier du mois correspondant. Soit i numéro du mois et j celui de l'année, nous avons donc :

$$Xcvs_t = Xcvs_{i,j} = X_{i,j} - c'_i$$

Nous réalisons ensuite une régression linéaire sur $Xcvs_t$ afin d'estimer la tendance sous la forme suivante : $m(t) = \beta_1 + \beta_2 \cdot t$. Nous obtenons les coefficients β_1 et β_2 grâce à la méthode des moindres carrés ordinaires.

$$\text{Soit } M = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ \vdots & \vdots \\ 1 & 216 \end{pmatrix} \text{ et } V = \begin{pmatrix} Xcvs_1 \\ Xcvs_2 \\ \vdots \\ Xcvs_{216} \end{pmatrix}$$

$$\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = (M^T \cdot M)^{-1} \cdot (M^T \cdot V) = \begin{pmatrix} 16.928 \\ 0.0096 \end{pmatrix} \quad (1)$$

Nous obtenons $\beta_1 = 16.928$ et $\beta_2 = 0.0096$. Nous pouvons ensuite construire les valeurs de la série corrigée des variations saisonnières sur l'ensemble de prédiction :

$$X'_{cvs}(t) = \beta_1 + \beta_2 \cdot t$$

Il suffit alors d'ajouter le coefficient saisonnier correspondant aux valeurs X'_{cvs} prédites. C'est à dire que si on considère le couple (i, j) où i est le numéro du mois et j celui de l'année, on a :

$$X'(i, j) = X'_{cvs}(i, j) + c'_i$$

Nous calculons ainsi les prédictions de janvier 2018 à février 2023.

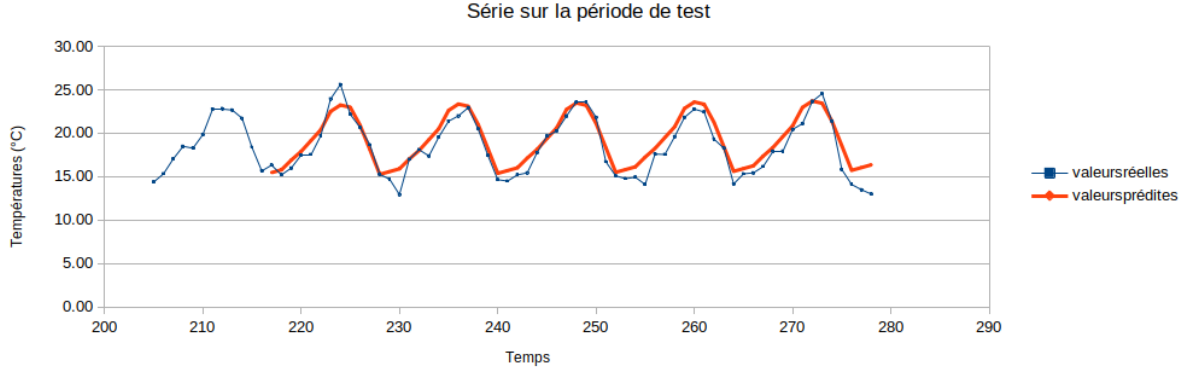


FIGURE 9 – Série obtenue en ajoutant les coefficients saisonniers trouvés par moyenne mobile à la tendance prédite.

Afin d'évaluer la qualité de notre modèle, nous calculons l'erreur absolue moyenne sur l'ensemble de test, différence entre les valeurs prédites et les valeurs réelles. On obtient alors $MAE = 1.0397$, ce qui signifie que l'écart moyen entre les températures réelles et les températures prédites est de 1.0397 degrés Celsius.

2.2 Régression sur le temps avec des variables indicatrices

La méthode de régression sur le temps permet d'estimer à la fois la tendance et les coefficients saisonniers. Nous allons utiliser un modèle sans constante avec comme variables explicatives le temps ainsi que 12 variables indicatrices (une pour chaque mois). La constante β_1 sera alors la moyenne des coefficients saisonniers provisoires trouvés (qu'on centrera ensuite). On écrira :

$$X_t = \beta_2 \cdot t + \sum_{k=1}^{12} c_i \cdot 1_k(i)$$

avec c_i le coefficient saisonnier correspondant au numéro du mois i ($t = i + 12 \cdot (j-1)$). Nous mettons en effet

Notons M_i la matrice à 13 colonnes des variables explicatives et X_t les températures moyennes mensuelles des données d'entraînement. On calcule donc les coefficients par la méthode des moindres carrés :

$$\begin{pmatrix} \beta_2 \\ c_1 \\ c_2 \\ \vdots \\ c_{12} \end{pmatrix} = (M^T \cdot M)^{-1} \cdot (M^T \cdot V) \text{ avec } V = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_{216} \end{pmatrix}$$

Nous trouvons alors $\beta_1 = 16.927$ (moyenne des coefficients saisonniers trouvés) et $\beta_2 = 0.0096$. On centre alors les coefficients saisonniers en leur retirant β_1 .

	c_i	c'_i
Janvier	13.423	-3.504
Février	13.735	-3.192
Mars	14.750	-2.177
Avril	15.811	-1.116
Mai	17.052	0.124
Juin	18.330	1.402
Juillet	20.384	3.457
Août	21.063	4.136
Septembre	20.798	3.870
Octobre	18.775	1.848
Novembre	15.923	-1.004
Décembre	13.083	-3.844

On calcule ensuite les valeurs de prévisions de la même façon que pour la moyenne mobile en calculant d'abord $X'_{cvs}(t) = \beta_1 + t \cdot \beta_2$ avec $t \in \llbracket 217; 278 \rrbracket$ puis en ajoutant les coefficients saisonniers correspondants.

On trouve une erreur absolue moyenne de 1.045 sur l'ensemble d'entraînement.

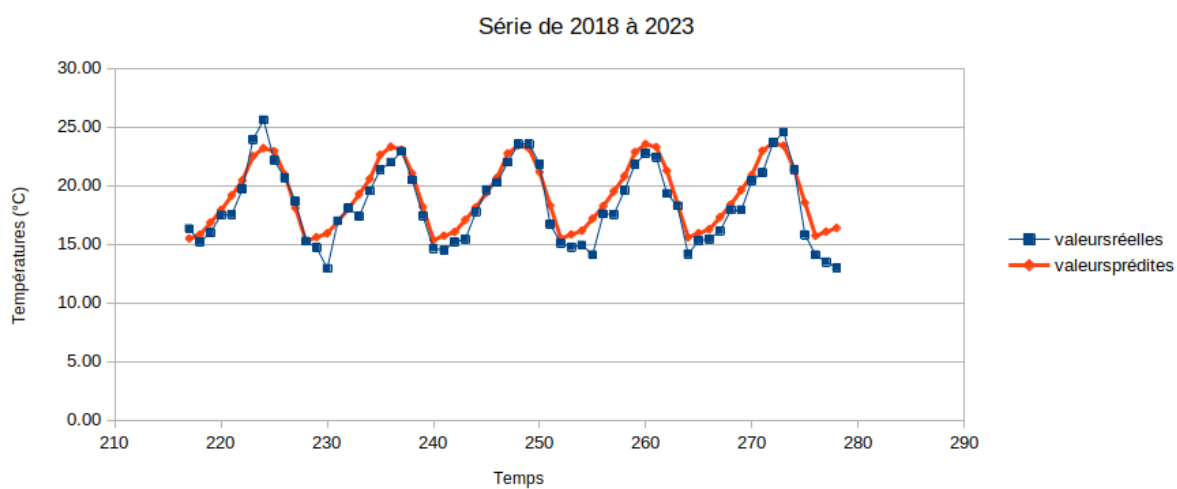


FIGURE 10 – Prévisions obtenues par la méthode de régression linéaire avec variables indicatrices.

3 Prévisions par lissage exponentiel

Dans cette partie, nous utiliserons deux méthodes de lissage exponentiel pour prédire les valeurs de la série sur l'ensemble de test.

3.1 Lissage exponentiel simple

La méthode de lissage exponentiel simple (LES) consiste en le calcul d'une moyenne pondérée de la série temporelle en donnant plus de poids aux observations les plus récentes. Elle ne prend pas en compte les tendances et les saisonnalités et retourne une valeur constante (peu importe l'horizon de prévision). Pour décider de l'importance donnée aux valeurs anciennes, nous utilisons une constante de lissage α comprise entre 0 et 1. Les formules qui suivent permettent de calculer le lissage exponentiel simple de la série sur la base de l'observation n à l'horizon h .

$$\begin{aligned}\forall h \quad \hat{x}_{n,h} &= \alpha \sum_{j=0}^{n-1} (1-\alpha)^j x_{n-j} \\ &= \alpha x_n + (1-\alpha) \hat{x}_{n-1,h}\end{aligned}\tag{2}$$

Plus le coefficient de lissage α est petit, plus la méthode donnera de l'importance aux observations anciennes et inversement. Nous avons alors testé différentes valeurs pour α (0.1, 0.2, ..., 0.8, 0.9, 1) et nous avons constaté que les prévisions étaient meilleures lorsqu' α était petit (c'est à dire en donnant de l'importance aux données anciennes). Nous retenons donc la valeur $\alpha = 0.1$.

Ici, nous appliquerons la méthode à la série corrigée des variations saisonnières obtenues dans la partie précédente.

```
LES <- ets(X_cvs, model = "ANN", additive.only = TRUE, alpha = 0.1)
summary(LES)
test_pred <- forecast(LES, h = 62)
```

FIGURE 11 – Code R pour créer le modèle de LES et effectuer les prédictions.

Notons que "ANN" signifie que nous utilisons un modèle additif. Le résumé du modèle indique une erreur absolue moyenne de 0.08 degrés Celsius sur l'ensemble d'entraînement. La prédiction donne comme valeur constante prédite :

$$\forall h \quad x_{n,h} = 18.9105 \text{ degré Celsius}$$

En utilisant la formule récursive du LES sur Excel, nous obtenons exactement le même résultat. Nous ajoutons ensuite les coefficients saisonniers trouvés dans la partie précédente à la série et nous obtenons le graphique suivante :

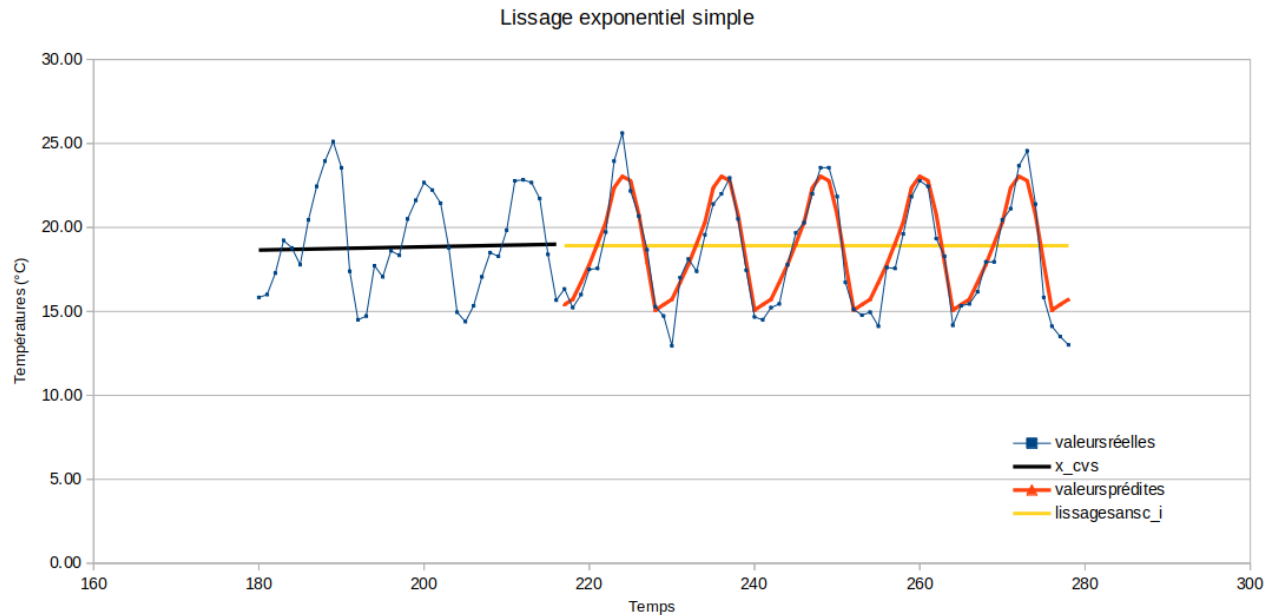


FIGURE 12 – Résultats obtenus pour le lissage exponentiel simple ($\alpha = 0.1$).

Sur la série test, nous obtenons une erreur absolue moyenne de 0.837 (sur R comme sur Excel), ce qui signifie que, en moyenne, l'écart entre la valeur réelle et la valeur prédite est de 0.837 degré Celsius.

3.2 Méthode de Holt Winters

La méthode de Holt Winters est une généralisation des lissages exponentiels simple et double et permet de prendre en compte la tendance et la saisonnalité des données. Elle utilise 3 paramètres de lissage, mais dont les valeurs sont plus complexes à calculer que dans le cas du LES. Nous utiliserons ici la fonction *HoltWinters()* de R, qui est capable de déterminer les paramètres optimaux. De plus, nous pouvons désormais entraîner le modèle sur la série initiale brute.


```

Call:
HoltWinters(x = train, alpha = NULL, beta = NULL, gamma = NULL,      seasonal = "add")

Smoothing parameters:
alpha: 0.3740256
beta : 0.006256784
gamma: 0.3025507

```

FIGURE 13 – Création du modèle de Holt Winters en R et paramètres de lissage.

Les paramètres de lissage utilisés et trouvés par R sont :

- $\alpha = 0.374$ pour la moyenne lissée,
- $\beta = 0.006$ pour la tendance lissée,
- $\gamma = 0.303$ pour la saisonnalité lissée.

Nous obtenons une tendance d'ordonnée à l'origine b et de coefficient directeur a , ainsi que les coefficients saisonniers dont les valeurs sont retournées par le modèle.

Coefficients:	
	[,1]
a	19.036028331
b	-0.007116345
s1	-3.452871698
s2	-2.617638895
s3	-1.665862404
s4	-0.762349358
s5	-0.194436463
s6	1.328458494
s7	3.594964077
s8	4.379758115
s9	4.555591747
s10	2.818833599
s11	-0.669051847
s12	-3.630427533

FIGURE 14 – Coefficients obtenus par le modèle de Holt Winters.

Remarquons que le modèle donne une tendance légèrement décroissante, à l'inverse de ce qu'on peut voir graphiquement et des résultats obtenus dans les parties précédentes.

Nous pouvons maintenant utiliser le modèle pour prédire les températures moyennes mensuelles sur la période de test. Encore une fois, nous prédisons à l'horizon 62 (nombre de mois dans l'ensemble de test).

```
HW_pred <- predict(HW_model, n.ahead = 62)
```

FIGURE 15 – Fonction R pour effectuer la prédiction avec le modèle de Holt Winters.

En calculant l'erreur absolue moyenne, nous trouvons que l'écart moyen entre les températures moyennes réelles et les températures moyennes prédites par le modèle de Holt Winters est d'environ 0.918 degré Celsius.



FIGURE 16 – Superposition des prédictions et de la série réelle sur l'ensemble de test.

3.3 Comparaison des deux méthodes

Graphiquement, aucune des deux méthodes (LES et Holt Winters) ne se distingue. En effet, les courbes reliant les valeurs de températures à chaque instant sur la période de test sont très similaires.

Le constat d'une tendance négative pour la méthode de Holt Winters et en se basant sur l'erreur absolue moyenne, la prédiction semble meilleur avec le lissage exponentiel simple. Cela peut paraître surprenant mais ici, le lissage exponentiel simple semble plus adapté à la série.

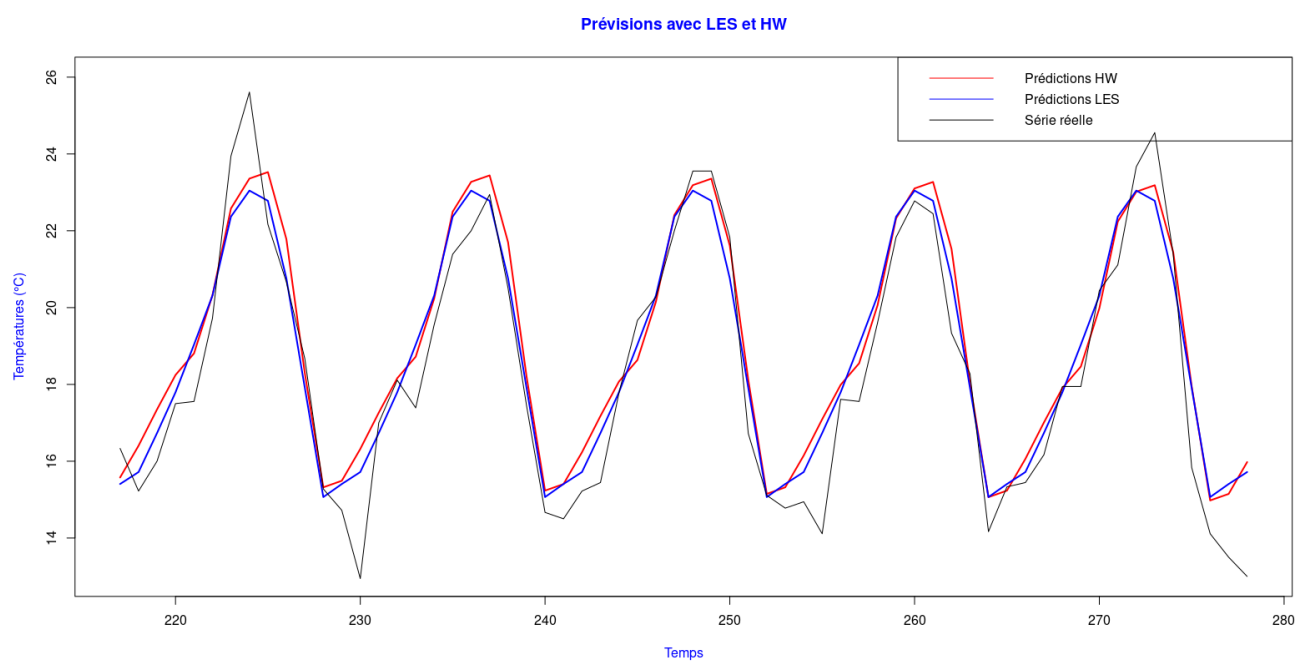


FIGURE 17 – Superposition des prédictions par les 2 méthodes sur l'ensemble de test.

Annexes : code R

```
# Importation des bibliothèques
library(ggplot2)
library(forecast)

##### Description des données #####

# Lecture des données
data <- read.table("donnees.csv", header = TRUE, sep = ",", skip = 0)

# Conversion des températures en °C
data$Value <- (data$Value - 32)/1.8

# Ajout de la colonne des indices temporelles
data["t"] <- 1:278

# Estimation de la température en fonction du temps par régression linéaire
# afin de dégager la tendance
model <- lm(Value ~ t, data = data)
summary(model)
# On trouve  $X = a + b \cdot t$  avec  $a = 17.167564$  et  $b = 0.006426$ 
# On constate une tendance croissance

# Affichage de la série et de la tendance
plot(Value ~ t, data = data, type = "l", main = "Série et tendance",
      xlab = "Temps", ylab = "Températures (°C)", col.lab = 'blue',
      col.main = 'blue')
abline(model, col = "red")

# Création de la série temporelle
serie <- ts(data$Value, start = c(2000, 1), end = c(2023, 2), frequency = 12)
print(serie)
plot(serie, main = "Séries des températures moyennes mensuelles à San Diego",
      xlab = "Temps", ylab = "Température (°C)", col.main = 'blue')

# Affichage des séries saisonnières
monthplot(serie)
```

```
# Affichage par années
colors <- rainbow(23)
plot(serie[1:12], type = "l", ylim = c(min(serie), max(serie)),
     xlab = "Mois", ylab = "Température (°C)")
for (i in 1:23) lines(serie[(1+12*i):(12*(i+1))], col = colors[i])
legend("topright", legend = paste("Année", 2000:2022), col = colors, lty = 1)

# Autocorrélation
auto_corr <- acf(serie, lag.max = 60, type = 'correlation', plot = T)
# On voit une saisonnalité de période 12
# Et confirmation d'une tendance (fonction d'auto-corrélation qui décroît)

# Nuages de points (X_t, X_(t+h)) pour h = 1, h = 3, h = 6 et h = 12
par(mfrow=c(2,2))
matrice_1 <- matrix(0, 277, 2)
cpt <- 1
for (i in 1:277){
  matrice_1[cpt, 1] <- data$Value[(i-1)*12 + 1]
  matrice_1[cpt, 2] <- data$Value[i*12 + 1]
  cpt <- cpt + 1
}
matrice_1 <- data.frame(matrice_1)
plot(matrice_1$X1, matrice_1$X2, main = 'Nuage de points pour h = 1',
     xlab = 'X_t', ylab = 'X_(t+1)', col.main = 'blue', col.lab = 'blue')
matrice_3 <- matrix(0, 273, 2)
cpt <- 1
for (t in 1:3) {
  for (i in 1:91){
    matrice_3[cpt, 1] <- data$Value[(i-1)*3 + t]
    matrice_3[cpt, 2] <- data$Value[i*3 + t]
    cpt <- cpt + 1
  }
}
matrice_3 <- data.frame(matrice_3)
plot(matrice_3$X1, matrice_3$X2, main = 'Nuage de points pour h = 3',
     xlab = 'X_t', ylab = 'X_(t+3)', col.main = 'blue', col.lab = 'blue')
```

```
matrice_6 <- matrix(0, 270, 2)
cpt <- 1
for (t in 1:6) {
  for (i in 1:45){
    matrice_6[cpt, 1] <- data$Value[(i-1)*3 + t]
    matrice_6[cpt, 2] <- data$Value[i*3 + t]
    cpt <- cpt + 1
  }
}
matrice_6 <- data.frame(matrice_6)
plot(matrice_6$X1, matrice_6$X2, main = 'Nuage de points pour h = 6',
      xlab = 'X_t', ylab = 'X_(t+6)', col.main = 'blue', col.lab = 'blue')
matrice_12 <- matrix(0, 264, 2)
cpt <- 1
for (t in 1:12) {
  for (i in 1:22){
    matrice_12[cpt, 1] <- data$Value[(i-1)*12 + t]
    matrice_12[cpt, 2] <- data$Value[i*12 + t]
    cpt <- cpt + 1
  }
}
matrice_12 <- data.frame(matrice_12)
plot(matrice_12$X1, matrice_12$X2, main = 'Nuage de points pour h = 12',
      xlab = 'X_t', ylab = 'X_(t+12)', col.main = 'blue', col.lab = 'blue')

# Séparation des données en données d'entraînement et de validation
train <- window(serie, start = c(2000, 1), end = c(2017, 12))
test <- window(serie, start = c(2018, 1), end = c(2023, 2))

##### Désaisonnalisation par moyenne mobile #####

# Filtre de moyenne mobile d'ordre 12
filtre <- rep(1/12, 11)
filtre <- c(1/24, filtre, 1/24)
M12 <- filter(train, filter = filtre, sides = 2)
ts.plot(train, M12) # Superposition
```

```
# Calcul des coefficients saisonniers
S <- train - M12
print(S)
c_temp <- tapply(S, cycle(S), mean, na.rm = TRUE)
print(c_temp) # On constate qu'on a bien les mêmes coefs que sur excel
c_tmp_mean <- mean(c_temp)
print(c_tmp_mean)
c = c_temp - c_tmp_mean # On centre les coefficients saisonniers
print(c)
print(mean(c)) # On voit que la moyenne des coefficients est nulle
print(sum(c)) # De plus, leur somme est nulle

# Voyons les coefficients saisonniers donnés par la fonction décompose
decomp_train <- decompose(train, type = "additive")
coefs_decomp <- decomp_train$seasonal[1:12]
print(coefs_decomp)
print(c - coefs_decomp) # La différence fait 0: ce sont les mêmes

# Calcul de la série corrigées des variations saisonnières (sur train)
Xcvs <- train - decomp_train$seasonal

# Ajustement d'une tendance sur la série corrigée des variations saisonnières
df_train <- data.frame(Xcvs)
df_train$t = 1:216
model1 <- lm(Xcvs ~ t, data = df_train)
summary(model1)
a0 <- model1$coefficients[1]
a1 <- model1$coefficients[2]
# On trouve Xcvs_t_pred = 16.929 + 0.0096*t comme sur Excel

# Prédiction sur l'ensemble de test
df_test <- data.frame(test)
df_test$t <- 217:278
df_test$coef_saisonniersMM <- c(rep(c, 5), c[1], c[2])
df_test$predMM <- a0 + a1*df_test$t + df_test$coef_saisonniersMM
```

```

# Affichage des prédictions sur l'ensemble de test
ts_test_pred <- ts(df_test, start = c(2018, 1), end = c(2023, 2),
                  frequency = 12)
plot(df_test$t, df_test$test, type = "l", col = "blue", xlab = "Temps",
     ylab = "Températures (°C)", col.lab = 'blue',
     main = "Série de test: janvier 2018 à février 2023", col.main = 'blue')
lines(df_test$t, df_test$predMM, col = "red", lwd = "4")
legend("topright",
     legend = c("Série initiale", "Série prédite par moyenne mobile"),
     col = c("blue", "red"), lty = 1)

##### Désaisonnalisation par régression linéaire #####
##### avec variables indicatrices #####

# Création des variables indicatrices
D <- data[1:216, c('t', 'Value')] # Récupération des données d'entraînement
tmp <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
D$C1 <- rep(tmp, 18)
D$C2 <- c(0, rep(tmp, 17), 1, rep(0, 10))
D$C3 <- c(0, 0, rep(tmp, 17), 1, rep(0, 9))
D$C4 <- c(rep(0, 3), rep(tmp, 17), 1, rep(0, 8))
D$C5 <- c(rep(0, 4), rep(tmp, 17), 1, rep(0, 7))
D$C6 <- c(rep(0, 5), rep(tmp, 17), 1, rep(0, 6))
D$C7 <- c(rep(0, 6), rep(tmp, 17), 1, rep(0, 5))
D$C8 <- c(rep(0, 7), rep(tmp, 17), 1, rep(0, 4))
D$C9 <- c(rep(0, 8), rep(tmp, 17), 1, rep(0, 3))
D$C10 <- c(rep(0, 9), rep(tmp, 17), 1, rep(0, 2))
D$C11 <- c(rep(0, 10), rep(tmp, 17), 1, 0)
D$C12 <- c(rep(0, 11), rep(tmp, 17), 1)

# Régression linéaire sans constante
model2 <- lm(formula = Value ~ t+C1+C2+C3+C4+C5+C6+C7+C8+C9+C10+C11+C12+0,
             data = D)
summary(model2)
print(model2$coefficients)
# Mêmes coefficients obtenus sur Excel
beta2 <- model2$coefficients[1] # coefficient directeur
coefs_prov <- model2$coefficients[2:13] # coefficients saisonniers provisoires

```



```
# Calcul de la constante beta1 = moyenne des c_i trouvés
beta1 <- mean(coefs_prov)
print(beta1)

# On centre les coefficients
coefs_sais <- coefs_prov - beta1
print(coefs_sais) # Mêmes coefficients que sur Excel
print(sum(coefs_sais)) # La somme des coefficients saisonniers est bien nulle

# Prédiction sur la période 2018-2023
df_test$coef_saisonniersRVI <- c(rep(coefs_sais, 5), coefs_sais[1],
                                coefs_sais[2])
df_test$predRVI <- a0 + a1*df_test$t + df_test$coef_saisonniersRVI

# Affichage des prédictions sur la période 2018-2023
plot(df_test$t, df_test$test, type = "l", col = "blue", xlab = "Temps",
     ylab = "Températures (°C)", col.lab = 'blue',
     main = "Série de test: janvier 2018 à février 2023", col.main = 'blue')
lines(df_test$t, df_test$predRVI, col = "yellow", lwd = "4")
legend("topright", legend = c("Série initiale", "Série prédite par RVI"),
     col = c("blue", "yellow"), lty = 1)

# Comparaison coefficients saisonniers MM et RVI
print(df_test$coef_saisonniersMM[1:12])
print(df_test$coef_saisonniersRVI[1:12])
# Les coefficients saisonniers trouvés sont quasiment les mêmes avec des
# variations de l'ordre de 10^(-2)
print(df_test$coef_saisonniersMM[1:12] - df_test$coef_saisonniersRVI[1:12])

# Affichage des prédictions avec les deux méthodes
plot(df_test$t, df_test$test, type = "l", col = "blue", xlab = "Temps",
     ylab = "Températures (°C)", col.lab = 'blue',
     main = "Série de test: janvier 2018 à février 2023", col.main = 'blue')
lines(df_test$t, df_test$predMM, col = "red", lwd = "4")
lines(df_test$t, df_test$predRVI, col = "yellow", lwd = "4")
legend("topright", legend = c("Série initiale", "Série prédite par MM",
                             "Série prédite par RVI"),
     col = c("blue", "red", "yellow"), lty = 1)
# On remarque que les séries prédites avec les deux méthodes sont presque
# identiques
```

```
##### Lissage exponentiel simple #####
```

```
X_cvs <- a0 + a1*(df_train$t)
```

```
# Ajustement d'un LES (lissage exponentiel simple)
```

```
LES <- ets(X_cvs, model = "ANN", additive.only = TRUE) # LES pour modèle additif  
summary(LES)
```

```
# R a trouvé alpha = 0.9999 comme constante de lissage la plus adapté
```

```
# Cela signifie qu'on ne donne presque aucune importance aux observations
```

```
# anciennes.
```

```
# Par défaut, alpha est ici estimé par maximum de vraisemblance
```

```
# Avec une valeur de alpha
```

```
LES <- ets(X_cvs, model = "ANN", additive.only = TRUE, alpha = 0.1)
```

```
summary(LES)
```

```
# Nous souhaitons obtenir de meilleurs prévisions en donnant plus d'importance
```

```
# aux valeurs passées donc en réduisant la valeur d'alpha.
```

```
# De plus on remarque que la prévision est la meilleure en terme d'erreur
```

```
# absolue moyenne pour alpha = 0.1
```

```
# Prévisions sur l'ensemble de test (sans saisonnalités)
```

```
test_pred <- forecast(LES, h = 62)
```

```
print(test_pred)
```

```
plot(test_pred, main = 'Lissage exponentiel simple', col.main = "blue",  
      xlab = "Temps", ylab = "Températures (°C)", col.lab = 'blue', lwd = 2)
```

```
# Dans le cas du LES, la prévision est constante, 18.91 ici.
```

```
# On peut voir les intervalles de confiance à 80% et 95%
```

```
# On obtient des résultats plus proches de la réalité en ajoutant les
```

```
# coefficients saisonniers trouvés précédemment à la prédiction
```

```
# Ajout des coefficients saisonniers aux prédictions
```

```
plot(x = df_test$t, y = test_pred$mean + df_test$coef_saisonniersRVI,  
      type = 'l', main = 'Prévisions', col = 'red', ylim = c(13,26),  
      col.main = 'blue', xlab = 'Temps', ylab = 'Températures (°C)', lwd = 2,  
      col.lab = 'blue')
```

```
points(x = df_test$t, y = df_test$test, type = 'l')
```

```
points(x = df_test$t, y = rep(test_pred$mean[1], 62), type = 'l', col = 'blue')
```

```
legend("topright", legend = c("Prédictions", "Série réelle", 'LES sans saison'),  
      col = c("red", "black", 'blue'), lty = 1)
```

```
# Calcul de l'erreur absolue moyenne
eps_abs <- rep(0, 62)
for (i in 1:62) eps_abs[i] = abs(df_test$test[i] - (test_pred$mean[1] +
                                                    df_test$coef_saisonniersRVI[i]))

MAE_LES <- mean(eps_abs)
print(MAE_LES)
# L'ecart moyen entre les valeurs réelles et les valeurs prédite est d'environ
# 0.84

##### Lissage Holt Winters #####

# Création du modèle de Holt Winters
HW_model <- HoltWinters(train, alpha = NULL, beta = NULL, gamma = NULL,
                      seasonal = "add")
print(HW_model$coefficients)
# Les coefficients ressemblent à ceux trouvés avec les autres méthodes
print(HW_model)
# R trouve alpha = 0.37, beta = 0.006 et gamma = 0.30

# Prédiction sur l'ensemble de test
HW_pred <- predict(HW_model, n.ahead = 62)

# Affichage des prédictions
plot(x = df_test$t, y = HW_pred, type = 'l', main = 'Prévisions Holt Winters'
     , col = 'red', ylim = c(13,26), col.main = 'blue', xlab = 'Temps',
     , ylab = 'Températures (°C)', lwd = 2, col.lab = 'blue')
points(x = df_test$t, y = df_test$test, type = 'l')
legend("topright", legend = c("Prédictions HW", "Série réelle"),
      col = c("red", "black"), lty = 1)

# Calcul de l'erreur absolue moyenne
eps_abs_HW <- rep(0, 62)
for (i in 1:62) eps_abs_HW[i] = abs(df_test$test[i] - HW_pred[i])
MAE_HW <- mean(eps_abs_HW)
print(MAE_HW)
# L'ecart moyen entre les valeurs réelles et les valeurs prédites est d'environ
# 0.92.
```

```
# Affichage des deux prédictions: par LES et par HW
plot(x = df_test$t, y = HW_pred, type = 'l', main = 'Prévisions avec LES et HW'
     , col = 'red', ylim = c(13,26), col.main = 'blue', xlab = 'Temps',
     , ylab = 'Températures (°C)', lwd = 2, col.lab = 'blue')
points(x = df_test$t, y = test_pred$mean + df_test$coef_saisonniersRVI,
       type = 'l', col = 'blue', lwd = 2)
points(x = df_test$t, y = df_test$test, type = 'l', col = 'black')
legend("topright",
      legend = c("Prédictions HW", "Prédictions LES", "Série réelle"),
      col = c("red", "blue", "black"), lty = 1)
# A vue d'oeil, les deux prédictions sont assez similaires et on ne discerne pas
# de différence quant à leur efficacité

# Comparaison des erreurs absolues moyennes
print(MAE_LES)
print(MAE_HW)
# Elle est légèrement supérieure pour le LES que pour la méthode de HW
```