



ING 2

RAPPORT

Projet Séries Temporelles

Élèves :

Ayoub EL HOUDRI
Marine RASOLOFO

Enseignant :

M. Naamane LAÏB

25 mars 2022

Table des matières

Partie 1 - Caractéristiques de la série	2
1. Emplacement des données	2
2. Caractéristiques : modèle, tendance, saisonnalité	2
Partie 2 - Désaisonnalisation	4
Partie 3 - Prévision par lissage exponentiel	9
1. Lissage exponentiel simple	9
2. Lissage exponentiel de Holt-Winters	14
3. Comparaison des méthodes	16

Partie 1 - Caractéristiques de la série

1. Emplacement des données

Nous avons choisi une série qui modélise la production mensuelle de bière en Australie, entre janvier 1974 et août 1995 (476 observations au total).

Le jeu de données initial démarre à janvier 1956 mais nous avons réduit le nombre d'observations pour n'en garder que 84 (de janvier 1974 à décembre 1980), ce qui nous semblait suffisant, et rendait le travail un peu moins fastidieux (notamment pour le calcul des coefficients saisonniers sur tableur par exemple).

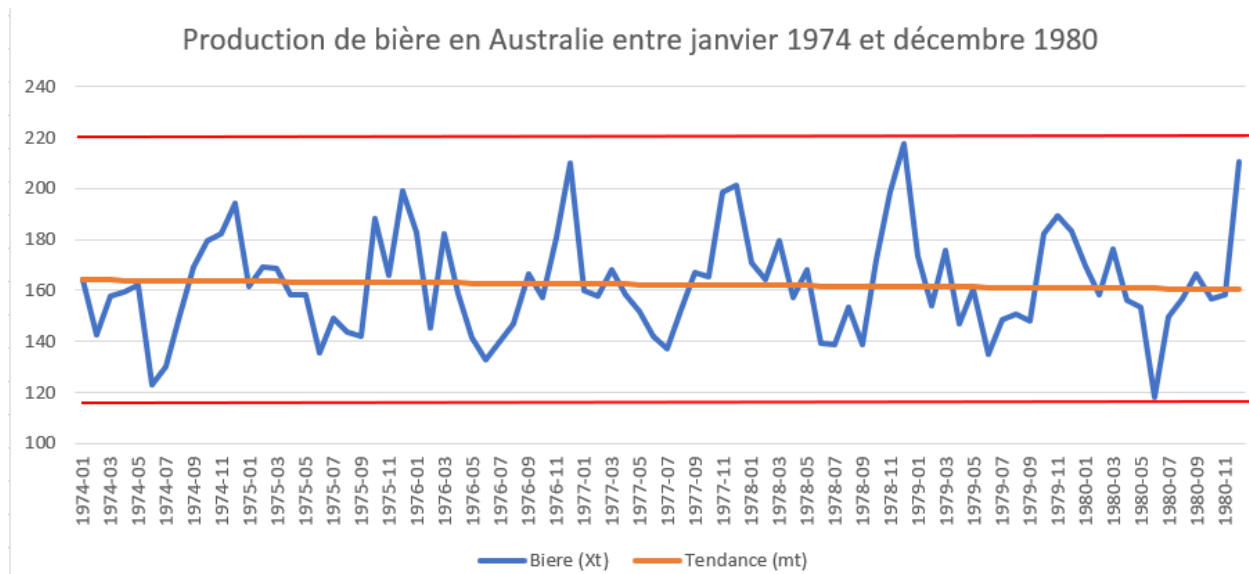
Vous pourrez trouver le jeu de données initial en cliquant sur **ce lien**, dans le dossier "1 Dataset", fichier "beer.csv".

Dans notre archive de rendu, vous trouverez nos données au format csv dans le dossier "Données". Le jeu de données initial est au nom de "beer.csv" et notre jeu de données "réduit" au nom de "beer_projet.csv".

2. Caractéristiques : modèle, tendance, saisonnalité

Nous observons :

- Un modèle additif : en traçant deux droites au-dessus et au-dessous des valeurs de la série, nous remarquons qu'elles sont parallèles
- Une tendance globalement constante, en très légère baisse
- Une saisonnalité, avec un pic de production de bière en période de fin d'année (décembre), et une baisse en début d'été (juin). Nous avons globalement une saisonnalité de période 12, autrement dit mensuelle.



Désormais, nous allons séparer cette série en 2 : 80% des observations (67 observations) seront considérées comme une série d'application, et les 20% restant (17 observations) serviront à valider nos prévisions.

Vous trouverez ces deux séries dans le dossier "Données", aux noms respectifs de "beer_projet_app.csv" et "beer_projet_test.csv"

Partie 2 - Désaisonnalisation

Nous travaillons donc maintenant avec les données "beer_projet_app" qui est constituée de 80% des observations de la série initiale.

Dans un premier temps, nous calculons la **moyenne mobile d'ordre 12**. Pour cela, nous calculons la moyenne des 12 premières valeurs de production de bière, puis nous étirons la cellule.

Ensuite, nous calculons la **série détendancialisée** en enlevant la moyenne mobile à l'observation correspondante.

Nous calculons les **coefficients saisonniers** de chaque mois. Pour cela, nous calculons la moyenne des valeurs de la série détendancialisée pour chaque mois. Nous obtenons les coefficients saisonniers suivants :

Calcul des coefficients saisonniers	
Janvier	6.87
Février	-4.85
Mars	11.61
Avril	-7.21
Mai	-6.96
Juin	-26.32
Juillet	-20.85
Aout	-14.26
Septembre	-9.10
Octobre	8.21
Novembre	23.00
Décembre	41.97

La somme de ces coefficients n'est pas égale à 0. Nous allons donc les centrer en enlevant la moyenne de tous les coefficients à chacun d'eux. Nous obtenons les coefficients saisonniers suivants :

Coef saisonniers centrés	
Janvier	6.69
Février	-5.03
Mars	11.44
Avril	-7.38
Mai	-7.13
Juin	-26.49
Juillet	-21.03
Aout	-14.44
Septembre	-9.27
Octobre	8.03
Novembre	22.82
Décembre	41.79
<i>total</i>	<i>0.00</i>

Comparons nos résultats avec ceux de la fonction `decompose` de R. Pour calculer les coefficients saisonniers, nous utilisons la fonction `decompose`, et nous prenons l'attribut "figure".

```
serie = read.table("beer_projet_app.csv", header = F, sep=",")
serie = serie[,2] # On garde seulement la colonne des valeurs
serie = ts(data = serie, start = c(1974,1), frequency = 12) # Transformation en série temporelle
# Coefficients saisonniers
CS = decompose(serie)$figure
```

Nous obtenons les coefficients saisonniers suivants :

```
> CS
[1]  5.853889 -4.536736 11.246597 -5.006528 -8.122153 -26.211736 -24.182778 -13.994444
[9] -6.753611  8.979722 21.676389 41.051389
```

Nous remarquons que les coefficients ont le même ordre de grandeur même s'ils ne sont pas identiques.

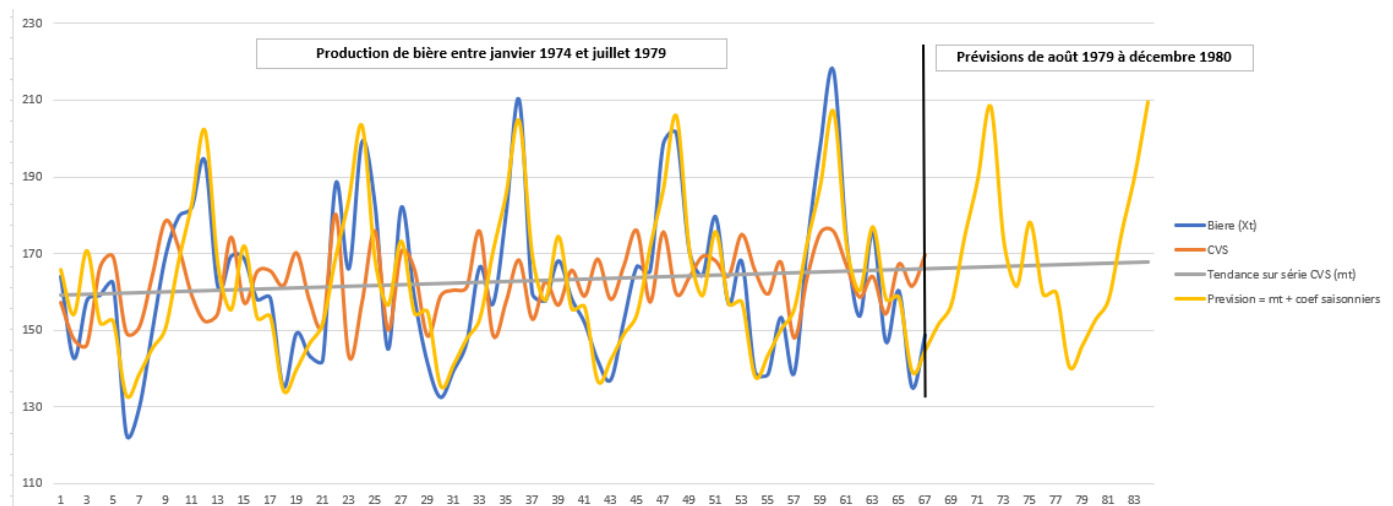
Nous avons ensuite calculé la **série désaisonnalisée (CVS)**, en enlevant les coefficients saisonniers à chaque observation (courbe orange sur le graphique suivant).

Ensuite, nous ajustons une **tendance par regression linéaire sur la série CVS**. Pour ce faire, nous calculons la variance et la moyenne de t , la moyenne de la série CVS et la covariance entre t et la série CVS. Puis nous calculons les coefficients de l'équation de la droite de régression. Nous obtenons l'équation : $mt = 159.03 - 0.1*t$. Cette droite est représentée par la courbe grise du graphique.

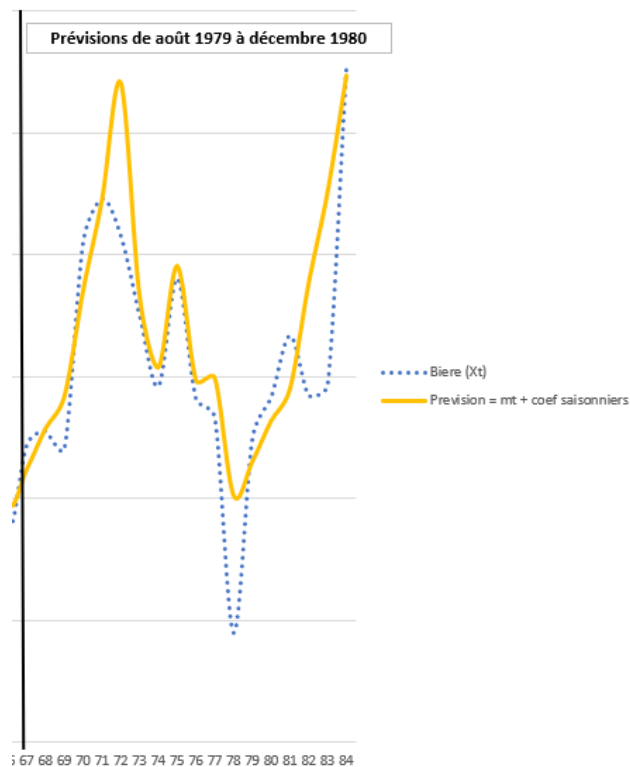
Calcul de la tendance sur la série CVS			
var(t)	379.667	a0	159.03
moy(t)	34	a1	0.10
moy(CVS)	162.53		
Cov(t, CVS)	39.0813	mt =159.03- 0.1*t	

L'étape suivante est la **prévision**. Pour cela, nous ajoutons les coefficients saisonniers à la tendance calculée précédemment. Nous obtenons la courbe jaune.

Voici le graphique regroupant toutes les étapes expliquées précédemment :

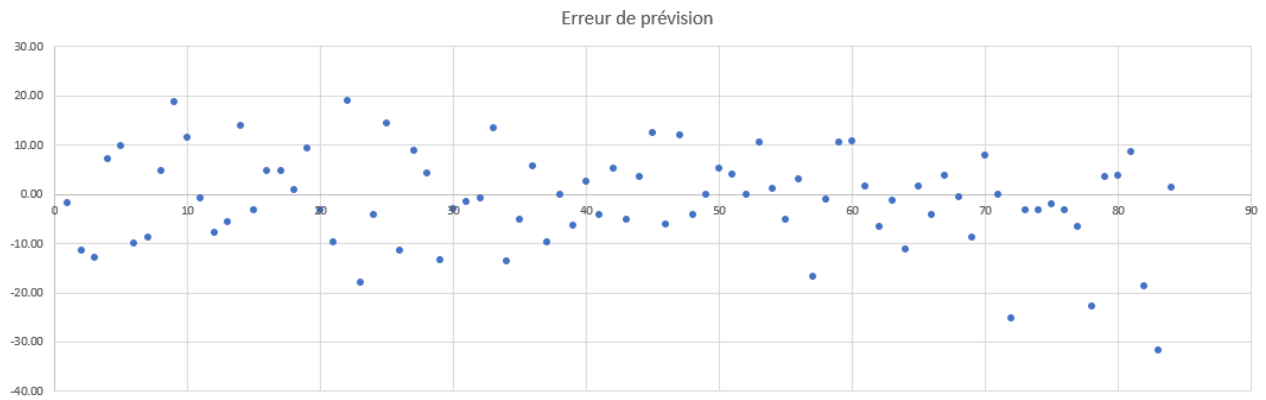


Tout d'abord, nous remarquons que la série prévisions (courbe jaune) suit la tendance et la saisonnalité de la série initiale (courbe bleue). Les prévisions ont l'air d'être convenables, mais pas excellentes. Les prévisions que nous avons effectuées sont faites à partir de la série "beer_projet_app", il est donc logique qu'elles soient convenables sur cette partie là. Les "vrais" prévisions sont donc réalisées sur la série "beer_projet_test". Nous allons donc pouvoir comparer notre prévision avec les vrais valeurs.



On remarque que la prévision n'est pas très bonne au niveau des "pics" de production, qu'il soit au plus haut ou au plus bas.

Pour avoir une idée plus précise, nous calculons les erreurs de prévision sur chaque observations, et nous obtenons le graphique suivant :



On remarque que les erreurs sont réparties à peu près uniformément autour de l'axe des abscisses, synonyme d'espérance nulle. De plus, on ne remarque pas de forme particulière, on peut supposer que la variance est constante, et donc que les erreurs ne sont pas corrélées. Cette interprétation nous permet de dire que la régression linéaire effectuée est considérée comme valide.

Partie 3 - Prédiction par lissage exponentiel

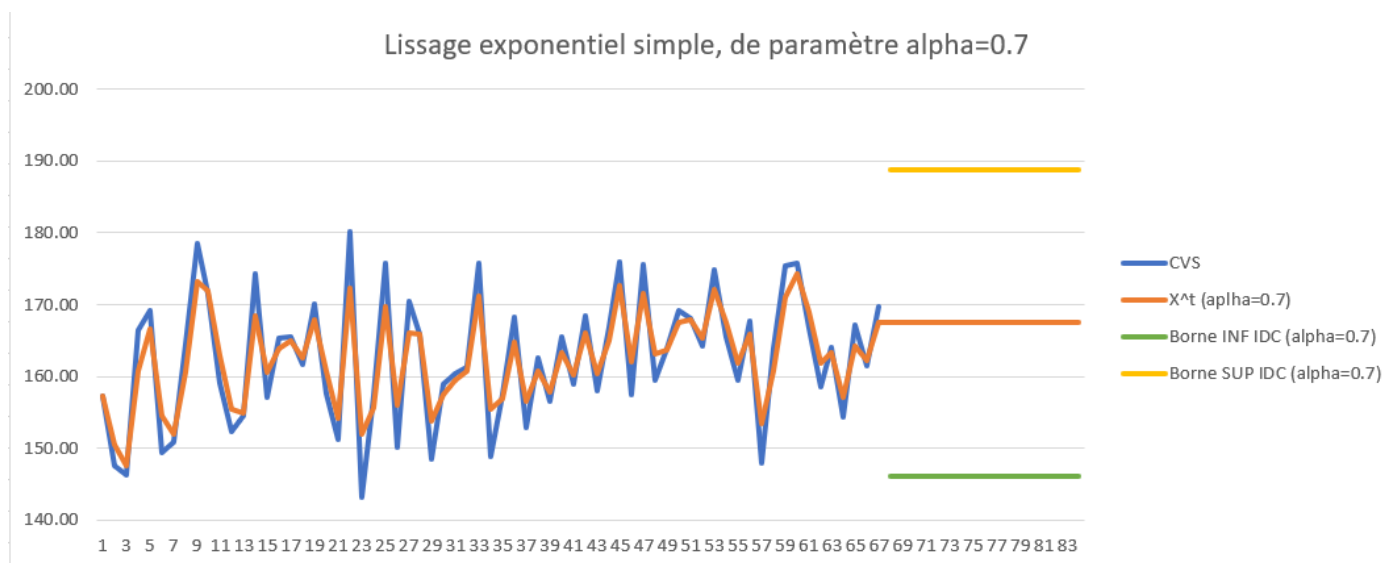
1. Lissage exponentiel simple

Nous avons le choix entre effectuer un lissage exponentiel simple ou double sur la série désaisonnalisée (CVS) car ce sont des méthodes qui s'appliquent à des séries sans composante saisonnières. Nous choisissons un **lissage exponentiel simple** car sur notre série CVS, nous observons que la tendance est en très très légère hausse, avec un écart-type égal à 2. Nous allons donc considéré que notre série CVS n'a pas de tendance.

Dans un premier temps, nous effectuons nos calculs sur tableur.

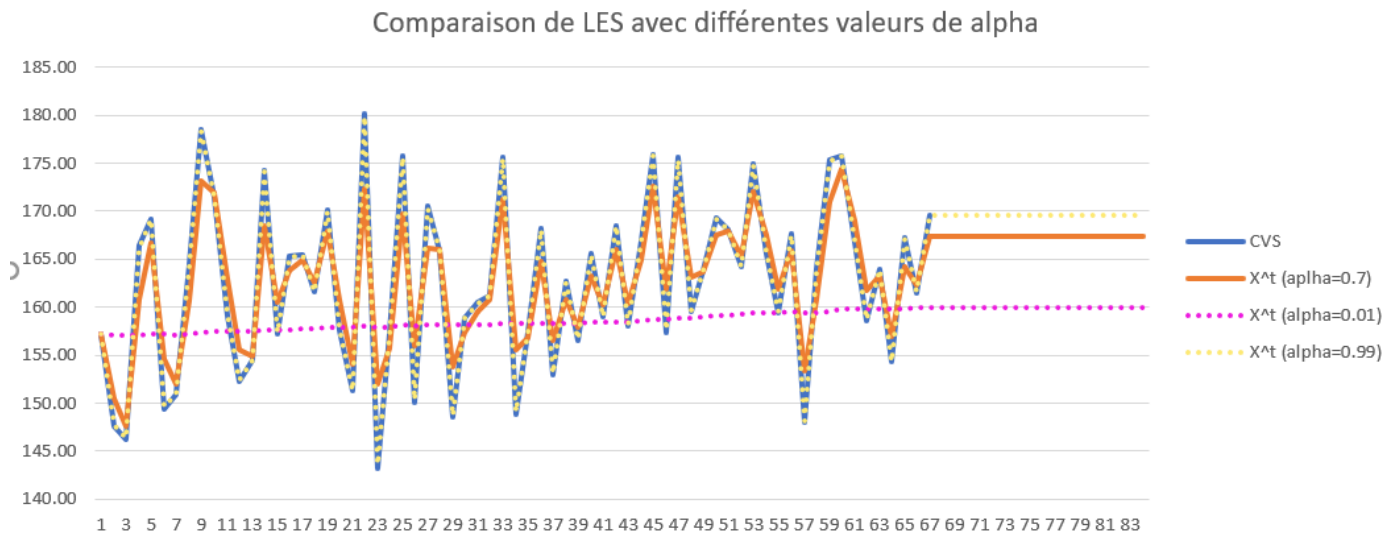
Nous choisissons un paramètre $\alpha=0.7$ initialement. Nous ferons varier ce paramètre pour voir ce qu'il implique.

Voici la prédiction que nous obtenons par lissage exponentiel simple, de paramètre $\alpha=0.7$. Pour l'initialisation, nous avons choisi de prendre l'observation au temps 1. Nous avons également calculé un intervalle de confiance à 95% pour les prévisions. Les bornes inférieure et supérieure sont représentées sur le graphiques respectivement en vert et jaune.



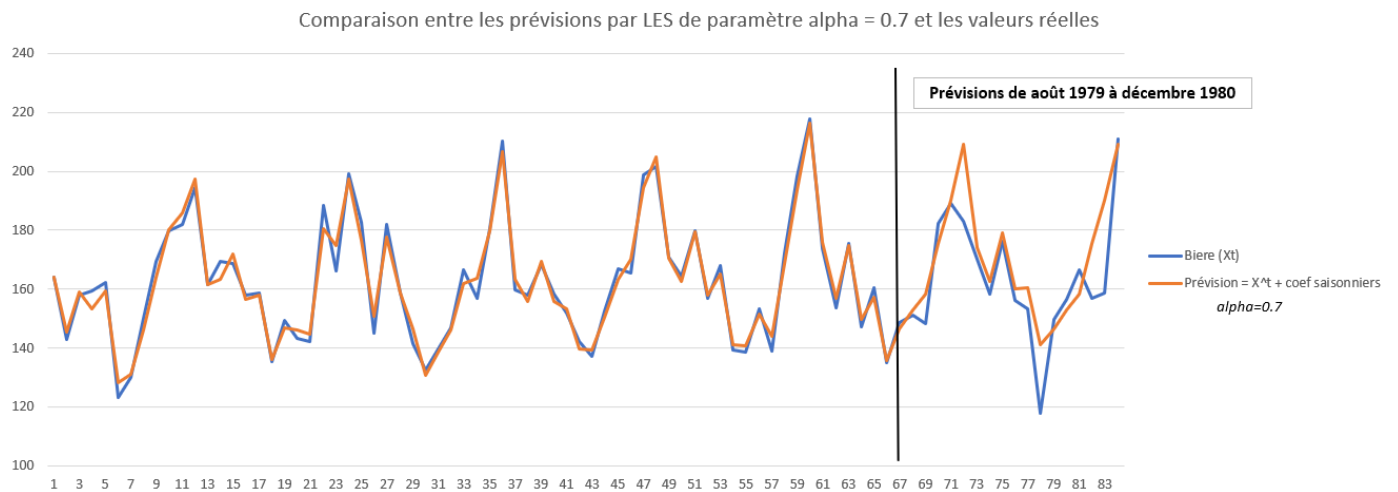
Nous remarquons que la courbe du lissage exponentiel simple suit globalement les variations de la courbe de la série CVS. Cela est dû au choix du paramètre α .

Faisons maintenant varier le paramètre alpha. Nous rajoutons sur le graphique les courbes représentant un lissage exponentiel simple de paramètre $\alpha=0.01$ (courbe pointillée rose) et $\alpha=0.99$ (courbe pointillée jaune) afin d'observer une réelle différence.



Nous savons que lorsque $\alpha=0.01$, la méthode prend en compte le passé lointain. Les prévisions sont peu sensibles aux variations. A l'inverse, lorsque $\alpha=0.99$, la méthode ne prend en compte que les valeurs récentes. Ainsi, les prévisions sont très sensibles aux fluctuations. Le graphique précédent illustre parfaitement ce principe puisqu'on observe la courbe rose en très légère hausse comparée à la toute première valeur de la série CVS (valeur la plus lointaine), alors que la courbe jaune suit exactement la courbe de la série CVS.

Nous allons maintenant conserver le lissage exponentiel de paramètre $\alpha=0.7$, et nous allons y ajouter les composantes saisonnières pour effectuer des **prévisions sur la série de test**. Nous obtenons les prévisions suivants (courbe orange) :

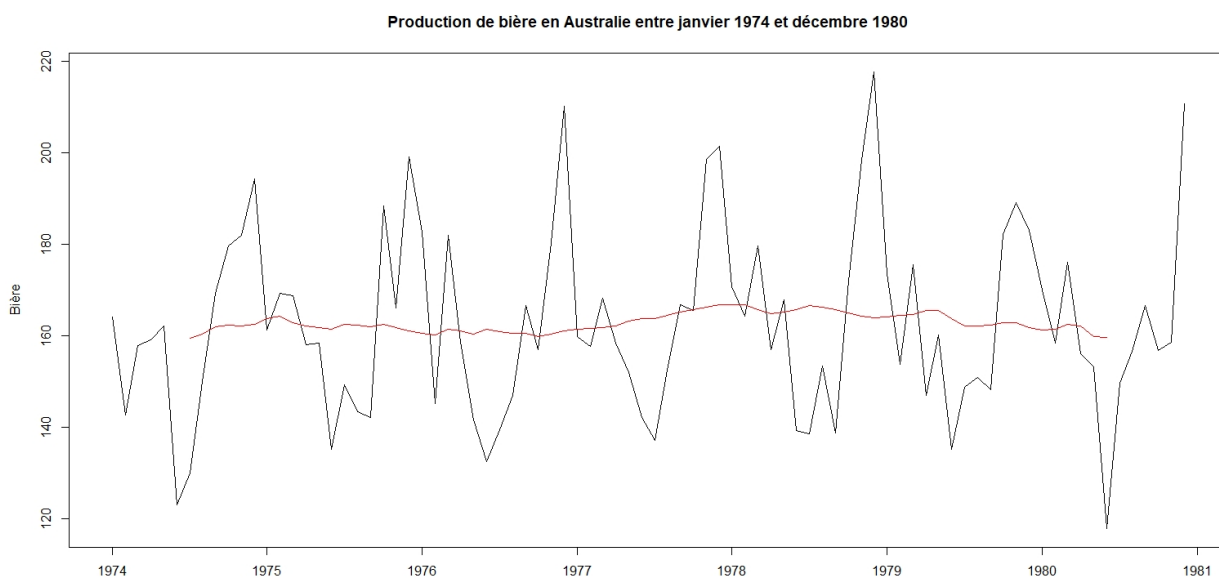


On remarque que la prévision n'est pas excellente mais convenable encore une fois.

Nous allons maintenant effectuer un lissage exponentiel simple **avec R**, puis nous allons comparer nos résultats avec ceux obtenus précédemment avec le tableur.

```
serie = read.table("beer_projet.csv", header = F, sep=",")
serie = serie[,2] # On garde seulement la colonne des valeurs
serie = ts(data = serie, start = c(1974,1), frequency = 12) # Transformation en série temporelle
serie.des = decompose(serie)$trend # série désaisonnalisée

# Affichage série et série désaisonnalisée
plot(serie, main="Production de bière en Australie entre janvier 1974 et décembre 1980", ylab="Bière", xlab="")
points(serie.des, type="l", col="red")
```



Tout d'abord, on remarque une première différence au niveau de la série désaisonnalisée. Les valeurs ne sont pas les mêmes que celles obtenues sur le tableur, et elles sont calculées entre juillet 1974 et janvier 1979.

On sépare la série désaisonnalisée en deux séries : application et validation, comme fait précédemment sur tableur.

```
# Séparation en deux séries : application et validation
serie.LES.app=window(serie.des,start=c(1974,1),end=c(1979,7))
serie.LES.val=window(serie.des,start=c(1979,8),end=c(1980,12))
```

Nous effectuons un lissage exponentiel simple à l'aide de la fonction `ets` du package `forecast`. Nous utilisons le modèle "ANN", qui signifie que l'on considère un modèle additif, sans tendance et sans saisonnalité.

```
# Lissage exponentiel simple sur la série application
library(forecast)
LES=ets(serie.LES.app,model="ANN",additive.only=TRUE)
summary(LES)
```

Nous obtenons les résultats suivants :

```
> summary(LES)
ETS(A,N,N)

Call:
ets(y = serie.LES.app, model = "ANN", additive.only = TRUE)

Smoothing parameters:
  alpha = 0.9999

Initial states:
  l = 159.4624

sigma: 0.7352

      AIC      AICc      BIC
217.2090 217.6300 223.5416

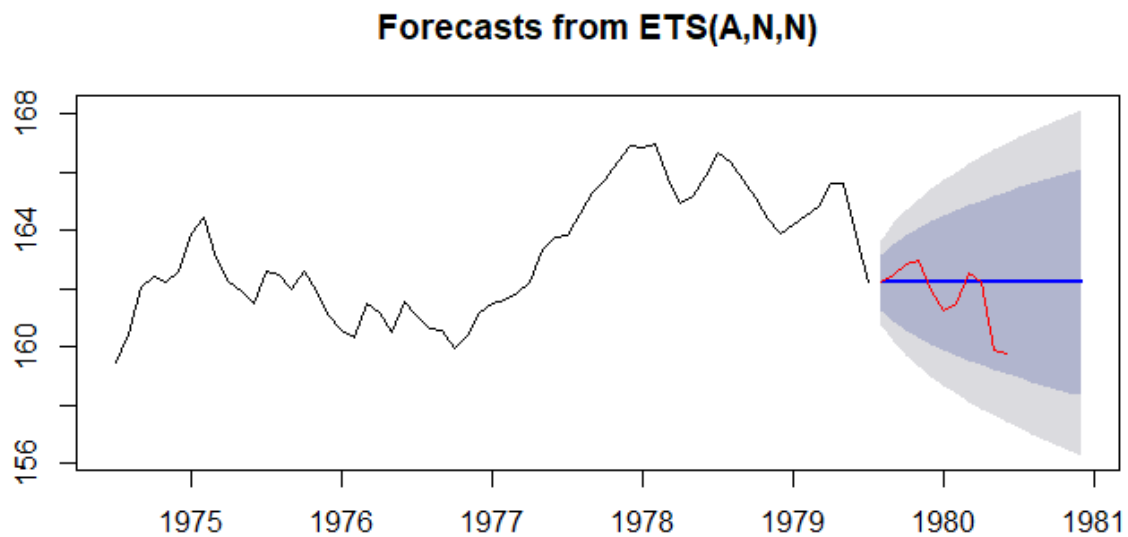
Training set error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
Training set 0.04481688 0.7230942 0.5960621 0.026884 0.3653724 0.2627025 0.3362487
```

On remarque que le paramètre `alpha` sélectionné par l'algorithme de R vaut 0.9999, ce qui rend le modèle très sensible aux fluctuations les plus récentes.

Enfin, nous effectuons des prévisions à l'aide de la fonction predict :

```
# Prévisions
prev=forecast(LES,h=17) # 17 mois dans la série validation
plot(prev)
points(serie.LES.val,type="l",col="red") # Comparaison avec série de validation en rouge
```

Nous obtenons la prévision suivante (courbe bleue) :



En comparant avec les valeurs de la série de validation (courbe rouge), on remarque que la prévision (courbe bleue) surestime les vraies valeurs de la série désaisonnalisée.

2. Lissage exponentiel de Holt-Winters

Pour effectuer un lissage exponentiel de Holt-Winters, nous utilisons la même commande que pour le lissage exponentiel simple en changeant le modèle que nous utilisons. Au lieu de "ANN" nous mettons "ANA". En effet, dans la série initiale, nous n'avons pas de tendance, on garde donc le "N" en deuxième position pour spécifier ceci, et on écrit "A" pour spécifier que l'on prend en compte la saisonnalité.

Voici notre code :

```
# Lissage exponentiel de Holt-winters
serie.HW.app=window(serie,start=c(1974,1),end=c(1979,7))
serie.HW.val=window(serie,start=c(1979,8),end=c(1980,12))

HW = ets(serie.HW.app,model="ANA",additive.only=TRUE)
summary(HW)

# Prévisions
prev=forecast(HW,h=17) # 17 mois dans la série validation
plot(prev)
points(serie.HW.val,type="l",col="red") |
```

Nous obtenons les résultats suivants :

```
> summary(HW)
ETS(A,N,A)

Call:
ets(y = serie.HW.app, model = "ANA", additive.only = TRUE)

Smoothing parameters:
  alpha = 1e-04
  gamma = 1e-04

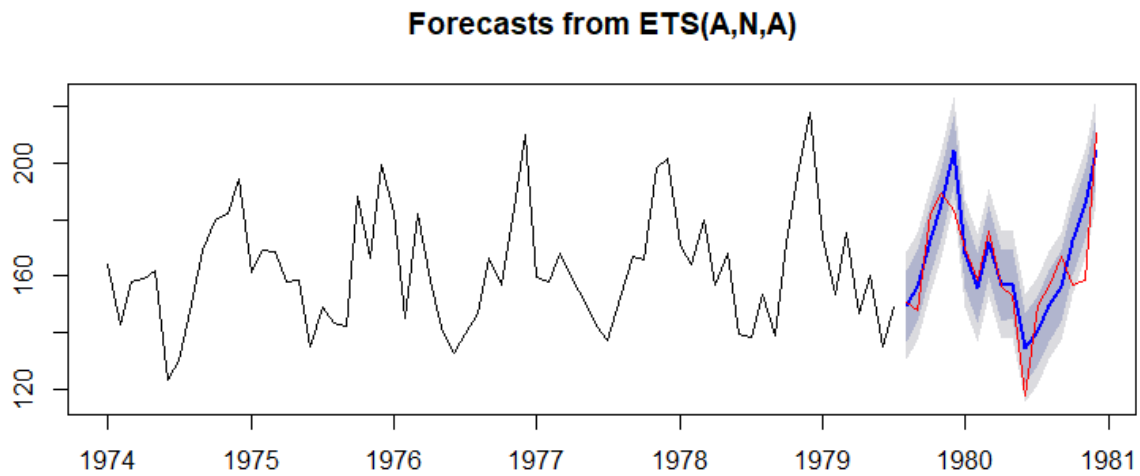
Initial states:
  l = 162.8373
  s = 41.6847 22.4912 9.692 -6.2259 -13.4433 -22.1738
      -28.2588 -5.3234 -6.1912 9.4384 -7.1008 5.4109

sigma: 9.7084

      AIC      AICc      BIC
600.5900 610.0018 633.6604

Training set error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
Training set -0.04885254 8.634695 7.043116 -0.315375 4.36369 0.6694977 -0.126686
```

Et enfin pour les prévisions :



En bleu, nous observons la courbe de prévision par lissage exponentiel de Holt-Winters. En rouge, ce sont les valeurs de la série de validation.

On remarque que les prévisions sont relativement proches des vraies valeurs. Par contre, les paramètres alpha et gamma sont très proches de 0.

3. Comparaison des méthodes

Tout d'abord, nous remarquons une différence au niveau du lissage exponentiel simple entre la version sur R et la version sur tableur. En effet, la prévision n'est pas tout à fait la même mais cela vient du fait que R n'utilise pas la méthode de moyenne mobile pour désaisonnaliser une série, alors que c'est cette méthode que nous avons utilisé sur tableur.

Ensuite, le lissage exponentiel de Holt-Winters donne des prévisions relativement proches des vraies valeurs.

On remarque que lorsqu'on a ajouté les coefficients saisonniers au lissage exponentiel simple sur tableur, nous avons obtenu la même courbe de prévision qu'avec la méthode Holt-Winters.

Finalement, on peut dire que la méthode que nous préférons est celle de Holt-Winters car elle est simple à implémenter sur R et donne des résultats fiables et satisfaisants.