

Projet Séries Temporelles Linéaires

MOUHLLI Rayane, VALOUR Thibaut

14 Mai 2021

1 Les données

1.1 Que représente la série choisie ?

Nous avons choisi une série représentant l'indice CVS-CJO de la fabrication de savons, de produits d'entretien et de parfums entre Janvier 1990 et Décembre 2012. La série comporte 276 données correspondant à la valeur de l'indice relevée chaque mois entre les deux dates données précédemment.

Notre série comporte une tendance linéaire croissante et ne semble pas stationnaire. Faisons une régression linéaire pour confirmer l'hypothèse faite sur la tendance.

```
Call:
lm(formula = savon ~ dates)

Residuals:
    Min       1Q   Median       3Q      Max
-19.3951  -2.4091   0.1651   2.8841  14.5544

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -7.164e+03  9.442e+01  -75.88  <2e-16 ***
dates        3.625e+00  4.717e-02   76.85  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.203 on 274 degrees of freedom
Multiple R-squared:  0.9557,    Adjusted R-squared:  0.9555
F-statistic: 5907 on 1 and 274 DF,  p-value: < 2.2e-16
```

Figure 1: Régression linéaire de la série $Savon_t$ sur le temps

Le coefficient β_{date} est bien positif (3.625) et significatif avec une p-value inférieure à 0.01. Le coefficient β_0 associé à la constante du modèle est négatif (-716,4) et est significatif avec une p-value inférieure à 0.01. Cette régression linéaire confirme bien notre hypothèse de tendance linéaire croissante.

1.2 On transforme la série pour la rendre stationnaire

Afin de rendre la série stationnaire, différencions la une fois. On définit donc la variable de différenciation comme suit

$$\Delta Savon_t = Savon_t - Savon_{t-1}$$

Nous allons réaliser différents tests afin de vérifier la stationnarité de notre nouvelle série ($\Delta Savon_t$). Puisque notre série temporelle comporte environ 270 données, ce qui est assez faible, réalisons deux des trois tests classiques de stationnarité afin de s'assurer de la pertinence du résultat.

Avant d'interpréter le test ADF, vérifions que les résidus du modèle de régression ne sont pas autocorrélés.

```
      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]     [,10]     [,11]     [,12]
lag  1.0000000 2.0000000 3.0000000 4.0000000 5.0000000 6.0000000 7.0000000 8.0000000 9.0000000 10.0000000 11.0000000 12.0000000
pval 0.3911778 0.1756832 0.1737963 0.2311733 0.316085 0.4089685 0.4473222 0.1920573 0.07718063 0.08540073 0.08492865 0.0995939

      [,13]      [,14]      [,15]      [,16]      [,17]      [,18]      [,19]      [,20]      [,21]      [,22]      [,23]      [,24]
lag 13.0000000 14.0000000 15.0000000 16.0000000 17.0000000 18.0000000 19.0000000 20.0000000 21.0000000 22.0000000 23.0000000 24
pval 0.1371679 0.1715881 0.19908 0.2422063 0.2625298 0.3065501 0.3613229 0.4228513 0.4689442 0.5297324 0.4237808 NA
```

Figure 2: Q-tests sur la série différencié $\Delta Savon_t$

On constate que l'absence d'autocorrélation n'est jamais rejetée à un seuil de 95%. Notre modèle est donc bien valide.

Ci-dessous, nous réalisons deux tests, le pp-test et l'ADF-test, qui consiste en le rejet de l'hypothèse de racine unitaire.

On constate que cette hypothèse est rejetée à un seuil de 95% pour la série différenciée ($\Delta Savon_t$) pour les deux tests. On peut donc conclure sur la viabilité de l'hypothèse de stationnarité.

```
Title:
Augmented Dickey-Fuller Test

Test Results:
PARAMETER:
Lag Order: 1
STATISTIC:
Dickey-Fuller: -13.3305
P VALUE:
0.01
```

Figure 3: ADF-test sur la série différencié ($\Delta Savon_t$)

```
Phillips-Perron Unit Root Test

data: dsavon
Dickey-Fuller Z(alpha) = -379.27, Truncation lag parameter = 5, p-value = 0.01
alternative hypothesis: stationary
```

Figure 4: PP-test sur la série différencié ($\Delta Savon_t$)

1.3 Représentons la série avant et après traitement

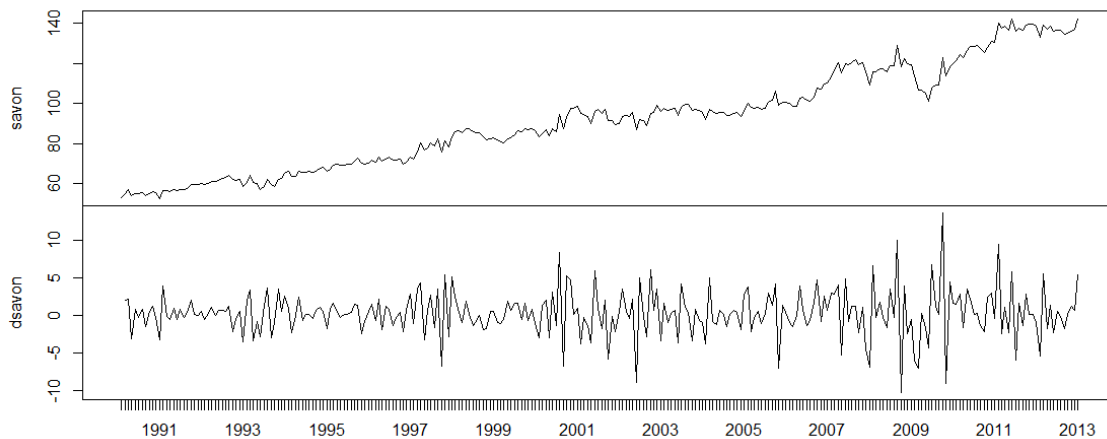


Figure 5: Représentation de la série ($Savon_t$) et de la série différenciée ($\Delta Savon_t$)

En fin de série, on observe une variance assez élevée. Une idée pourrait être de différencier une seconde fois, mais nous perdrons beaucoup trop d'informations. Nous préférons donc différencier seulement une fois ici pour rester le plus possible fidèle au modèle.

2 Modèles ARMA

2.1 Choisissons un modèle ARMA(p,q)

Affichons les autocorrélations totales et partielles afin de déterminer notre p_{max} et notre q_{max} pour notre modèle ARMA. En effet, nous pouvons les déterminer en regardant l'indice où les autocorrélations (totales et partielles) ne deviennent plus significativement non nulles à 5%.

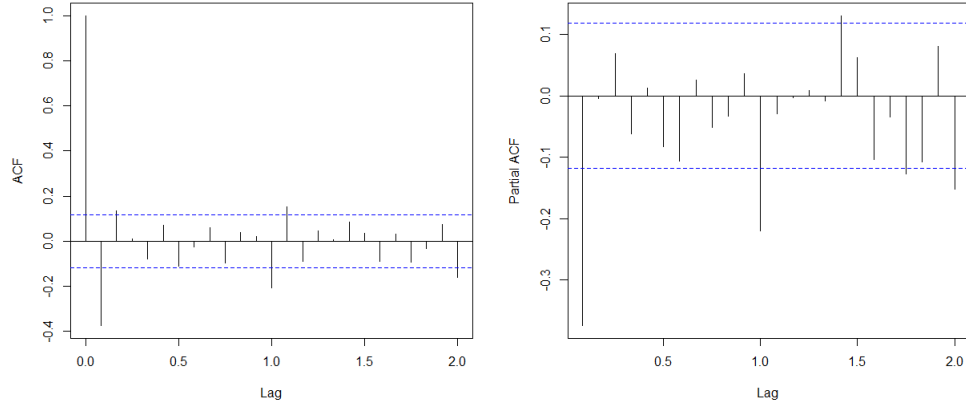


Figure 6: ACF et PACF sur la série différenciée ($\Delta Savont_t$)

L'étude de l'autocorrélogramme nous permet de choisir la valeur $q_{max} = 13$ (même s'il y a certains indices supérieurs à 13 qui sont significatifs, il est raisonnable de s'arrêter à 2 ici) et l'étude de l'autocorrélogramme partiel nous permet de choisir la valeur $p_{max} = 11$ (de même il y a certains indices significatifs au-delà de 11, mais il est raisonnable de s'arrêter à 11) .

Cherchons maintenant les différents modèles ARMA valides pour notre série temporelle ($\Delta Savont_t$)

	p	q	arsignif	masignif	resnocorr	ok
[1,]	6	3	1	1	1	1
[2,]	1	12	1	1	1	1
[3,]	0	13	NA	1	1	1

Figure 7: Différents modèles ARMA valides

Nous retenons trois modèles valides et ajustés. Calculons les AIC et BIC des modèles retenus afin de choisir ceux qui minimisent ces deux valeurs.

	arma(6,3)	arma(1,12)	arma(0,13)
AIC	1335.843	1335.260	1333.933
BIC	1375.667	1389.566	1388.239

Figure 8: AIC et BIC du modèle ARMA pour la série ($\Delta Savont_t$)

Deux couples sortent du lot : (6,3) qui minimise le BIC et (0,13) qui minimise l'AIC.

Regardons qui des deux modèles a le R^2 le plus important. En effet, celui qui a le plus grand R^2 a la meilleure prévision dans l'échantillon. On constate que l'ARMA(6,3) a un R^2 supérieur à celui de l'ARMA(0,13). On conserve donc ARMA(6,3) dans la suite.

```
> adj_r2(arma603)
[1] 0.2040029
> adj_r2(arma0013)
[1] 0.1937174
```

Figure 9: Comparaison du R^2 de l'ARMA(6,3) et de l'ARMA(0,13)

Tout d'abord, nous testons la nullité des coefficients et l'absence d'autocorrélation des résidus pour l'ARMA(6,3). Pour la significativité des coefficients, on peut vérifier que le rapport entre le coefficient

estimé et la variance estimée du coefficient estimé est bien supérieur en valeur absolue à 1.96. Exemple ci-dessous, sur le coefficient AR(6): $0.157/0.063 = 2.49 > 1.96$ et MA(3) : $0.953/0.045 = 21.17 > 1.96$.

De plus, le deuxième tableau nous indique que l'on peut rejeter l'hypothèse d'absence d'autocorrélation des résidus à 95%.

```

tests de nullité des coefficients :
      ar1  ar2  ar3  ar4  ar5  ar6  ma1  ma2  ma3 intercept
coef -0.341 0.004 0.90 0.274 -0.012 -0.157 -0.051 0.004 -0.953 0.305
se    0.069 0.068 0.07 0.063 0.065 0.063 0.039 0.029 0.045 0.016
pval  0.000 0.954 0.00 0.000 0.851 0.013 0.193 0.886 0.000 0.000

tests d'absence d'autocorrélation des résidus :
lag pval lag pval lag pval lag pval lag pval lag pval
[1,] 1 NA 7 NA 13 0.133 19 0.063 25 0.078 31 0.136
[2,] 2 NA 8 NA 14 0.197 20 0.063 26 0.106 32 0.124
[3,] 3 NA 9 NA 15 0.198 21 0.059 27 0.130 33 0.062
[4,] 4 NA 10 0.226 16 0.232 22 0.077 28 0.092 34 0.074
[5,] 5 NA 11 0.405 17 0.064 23 0.105 29 0.120 35 0.060
[6,] 6 NA 12 0.109 18 0.069 24 0.064 30 0.126 36 0.077

```

Figure 10: Test sur l'ARMA(6,3)

2.2 Modèle ARIMA(p,d,q)

A partir du modèle ARMA, établissons un modèle ARIMA(6,1,3) sur la série initiale ($Savon_t$). Nous fixons ici $d=1$ car nous avons travaillé précédemment sur une série différenciée 1 fois.

Vérifions la validité de notre modèle en vérifiant l'absence. d'autocorrélation des résidus.

```

      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
lag 1.0000000 2.0000000 3.0000000 4.0000000 5.0000000 6.0000000 7.0000000 8.0000000 9.0000000 10.0000000 11.0000000 12.0000000 13.0000000
pval 0.6390133 0.8547098 0.9179242 0.9710973 0.9695289 0.9800157 0.9169908 0.955214 0.971161 0.9833598 0.9769327 0.6326881 0.6720549
      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
lag 14.000000 15.000000 16.000000 17.000000 18.000000 19.000000 20.000000 21.000000 22.000000 23.000000 24.000000
pval 0.688026 0.703884 0.7453903 0.4798391 0.497578 0.41029 0.3632347 0.3083755 0.3361835 0.3919547 0.259326

```

Figure 11: Test d'autocorrélation des résidus pour l'ARIMA(6,1,3)

L'absence d'autocorrélation des résidus n'est jamais rejetée. L'ARIMA(6,1,3) est donc valide

3 Prévision

3.1 Equation vérifiée par la région de confiance niveau α sur les valeurs futures ($Savon_{T+1}$, $Savon_{T+2}$)

On a ici :

$$\begin{aligned}
\hat{Savon}_{T+1/T} &= \sum_{i=1}^6 \phi_i Savon_{T-(i-1)} + \sum_{i=1}^3 \psi_i \epsilon_{T-(i-1)} \\
\hat{Savon}_{T+2/T} &= \phi_1 \hat{Savon}_{T+1/T} + \sum_{i=2}^6 \phi_i Savon_{T-(i-2)} + \sum_{i=1}^3 \psi_i \epsilon_{T-(i-2)} \\
Savon_{T+1} &= \sum_{i=1}^6 \phi_i Savon_{T-(i-1)} + \sum_{i=1}^3 \psi_i \epsilon_{T-(i-1)} + \epsilon_{T+1} \\
Savon_{T+2} &= \sum_{i=1}^6 \phi_i Savon_{T-(i-2)} + \sum_{i=2}^3 \psi_i \epsilon_{T-(i-2)} + \epsilon_{T+2}
\end{aligned}$$

Calculons les erreurs de prédictions définies par $Savon_{T+1} - \hat{Savon}_{T+1/T}$ et $Savon_{T+2} - \hat{Savon}_{T+2/T}$. Matriciellement, le problème peut se réécrire comme suit

$$Savon - \hat{Savon} = \begin{pmatrix} Savon_{T+1} - \hat{Savon}_{T+1/T} \\ Savon_{T+2} - \hat{Savon}_{T+2/T} \end{pmatrix} = \begin{pmatrix} \epsilon_{T+1} \\ \epsilon_{T+2} + (\psi_1 + \phi_1) \epsilon_{T+1} \end{pmatrix} \quad (1)$$

Calculons maintenant la variance de $Savon - \hat{Savon}$. Par hypothèse de l'énoncé on a que les ϵ_t suivent une loi normale centrée de variance $\sigma_{\epsilon}^2 > 0$.

$$\begin{cases} V(Savon_{T+1} - \hat{Savon}_{T+1|T}) = V(\epsilon_{T+1}) = \sigma_\epsilon^2 \\ V(Savon_{T+2} - \hat{Savon}_{T+2|T}) = V(\epsilon_{T+2} + (\phi_1 + \psi_1)\epsilon_{T+1}) = \sigma_\epsilon^2(1 + (\psi_1 + \phi_1)^2) \end{cases} \quad (2)$$

Posons,

$$\Sigma = \begin{pmatrix} \sigma_\epsilon^2 & (\psi_1 + \phi_1)\sigma_\epsilon^2 \\ (\psi_1 + \phi_1)\sigma_\epsilon^2 & \sigma_\epsilon^2(1 + (\psi_1 + \phi_1)^2) \end{pmatrix} \quad (3)$$

Après calcul, nous trouvons que le déterminant de Σ est égal à $\sigma_\epsilon^4 > 0$, donc notre matrice de variance-covariance est inversible. D'après les hypothèses faites précédemment, $Savon - \hat{Savon} \sim \mathcal{N}(0, \Sigma)$. donc d'après le cours $\{(Savon - \hat{Savon})\Sigma^{-1t}(Savon - \hat{Savon}) \sim \chi^2(2)\}$.

Donc, en fixant $\alpha \in [0, 1]$, nous obtenons la région de confiance suivante : $\{(Savon - \hat{Savon})\Sigma^{-1t}(Savon - \hat{Savon}) \leq q_{1-\alpha}\}$

où $q_{1-\alpha}$ est le quantile d'ordre $1 - \alpha$ d'une loi $\chi^2(2)$ (χ^2 à 2 degrés de liberté).

3.2 Hypothèses

Les hypothèses utilisées pour obtenir cette région sont :

- Les bruits blancs sont des bruits blancs forts, c'est-à-dire une suite de variables aléatoires i.i.d suivant une loi normale centrée, de variance strictement positive.
- On suppose que le modèle est connu, c'est-à-dire que l'on connaît les coefficients de notre modèle ARIMA.
- On suppose que l'innovation linéaire $Savon_t - \hat{Savon}_t$ suit une loi normale centrée de variance strictement positive.

3.3 Représentation graphique

Tout d'abord représentons graphiquement la région pour la série $\Delta Savon_t$. La prédiction ne semble pas très précise mais l'intervalle de confiance est cohérent.

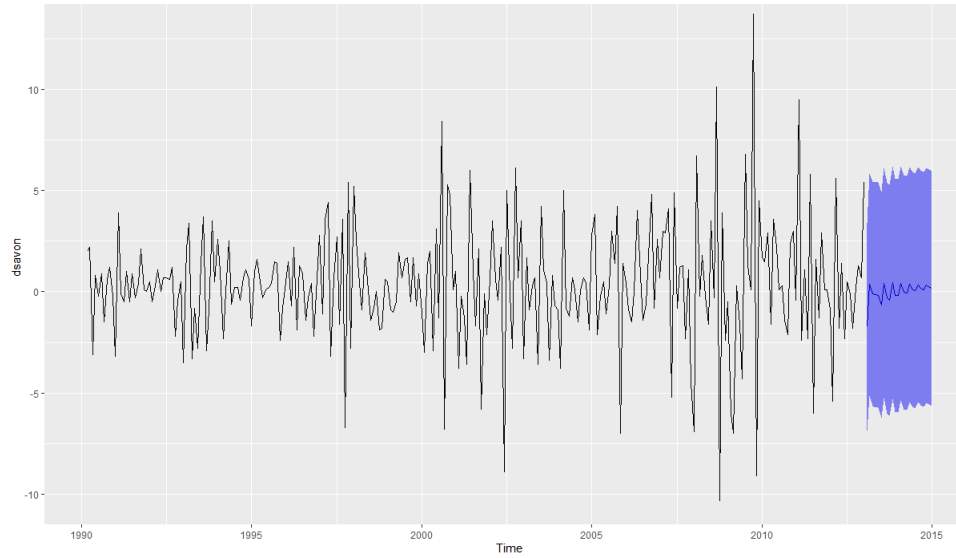


Figure 12: Estimation du modèle ARMA(6,3) à 95%

Ici, nous représentons graphiquement la région à 95% en bleu clair et à 80% pour la région en bleu foncé pour la série $Savon_t$. Nous pouvons conclure, de même, que la prédiction ne me semble pas très précise à cause de la faible variance mais l'intervalle de confiance semble cohérent.

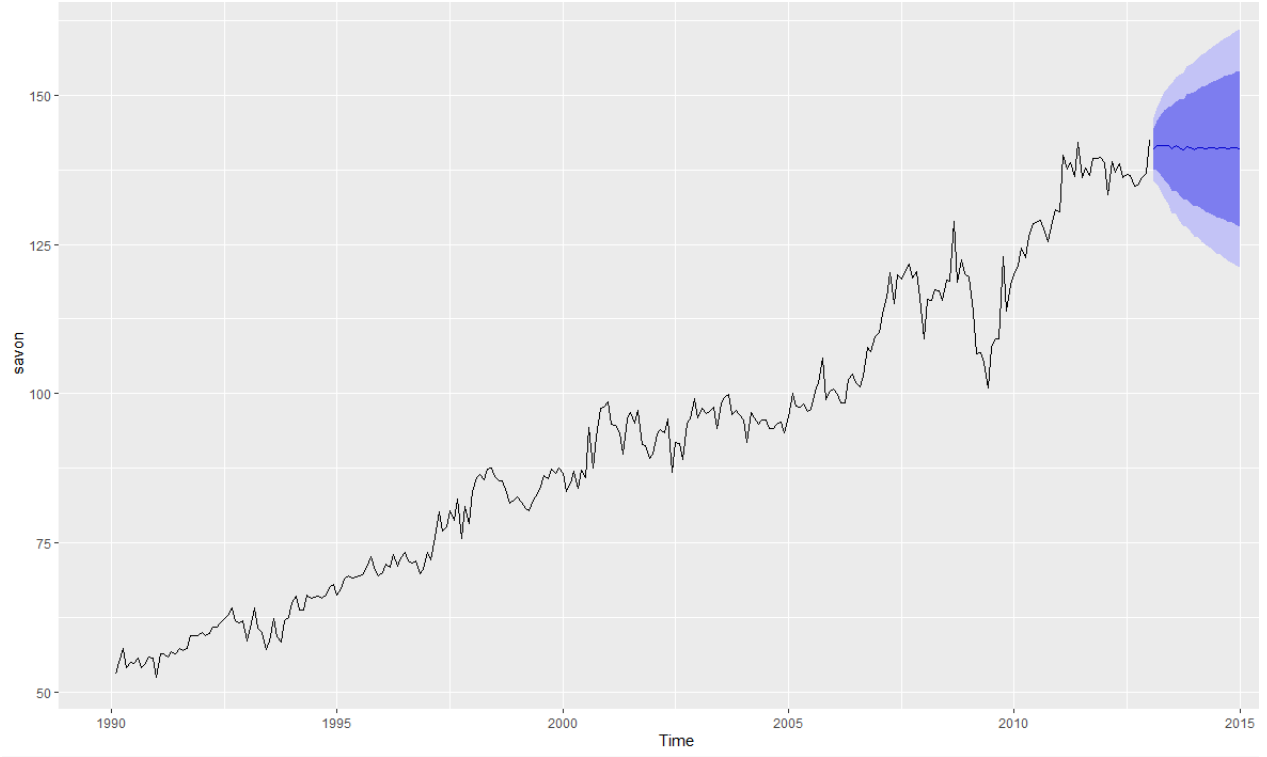


Figure 13: Estimation du modèle ARIMA(6,1,3) à 80% et 95%

3.4 Question ouverte

En supposant que Y_{T+1} soit disponible plus rapidement que X_{T+1} , on peut améliorer la prévision de X_{T+1} si (Y_t) cause instantanément (X_t) au sens de Granger. Le terme "cause" ne doit pas être interprété comme la causalité au sens usuel, mais plutôt comme l'utilité qu'a une variable pour en prédire une autre. Mathématiquement cela se traduit par

$$\hat{X}_{t+1}|\{X_u, Y_u, u \leq t\} \cup Y_{t+1} \neq \hat{X}_{t+1}|\{X_u, Y_u, u \leq t\}$$

Afin d'améliorer la prévision de X_{t+1} il faut donc que l'erreur quadratique sachant Y_{t+1} soit inférieure à celle dans le cas où on aurait pas cette information. Autrement dit

$$E[(X_{t+1} - \hat{X}_{t+1}|\{X_u, Y_u, u \leq t\} \cup Y_{t+1})(X_{t+1} - \hat{X}_{t+1}|\{X_u, Y_u, u \leq t\} \cup Y_{t+1})'] < E[(X_{t+1} - \hat{X}_{t+1}|\{X_u, Y_u, u \leq t\})(X_{t+1} - \hat{X}_{t+1}|\{X_u, Y_u, u \leq t\})']$$

4 Annexes

```

tests de nullité des coefficients :
      ma1 ma2 ma3 ma4 ma5 ma6 ma7 ma8 ma9 ma10 ma11 ma12 ma13 intercept
coef -0.355 0.099 -0.080 -0.029 0.062 -0.177 0.004 -0.021 -0.121 0.040 -0.017 -0.308 0.143 0.316
se    0.061 0.066 0.066 0.069 0.067 0.069 0.076 0.067 0.070 0.078 0.074 0.078 0.071 0.040
pval  0.000 0.133 0.227 0.678 0.354 0.010 0.960 0.758 0.085 0.611 0.817 0.000 0.044 0.000

tests d'absence d'autocorrélation des résidus :
lag pval lag pval lag pval lag pval lag pval
[1,] 1 NA 7 NA 13 NA 19 0.146 25 0.087 31 0.214
[2,] 2 NA 8 NA 14 0.067 20 0.153 26 0.109 32 0.200
[3,] 3 NA 9 NA 15 0.187 21 0.099 27 0.135 33 0.142
[4,] 4 NA 10 NA 16 0.267 22 0.100 28 0.132 34 0.131
[5,] 5 NA 11 NA 17 0.090 23 0.126 29 0.167 35 0.107
[6,] 6 NA 12 NA 18 0.136 24 0.065 30 0.192 36 0.135

```

Figure 14: Test du modèle ARMA(0,13) à 95%

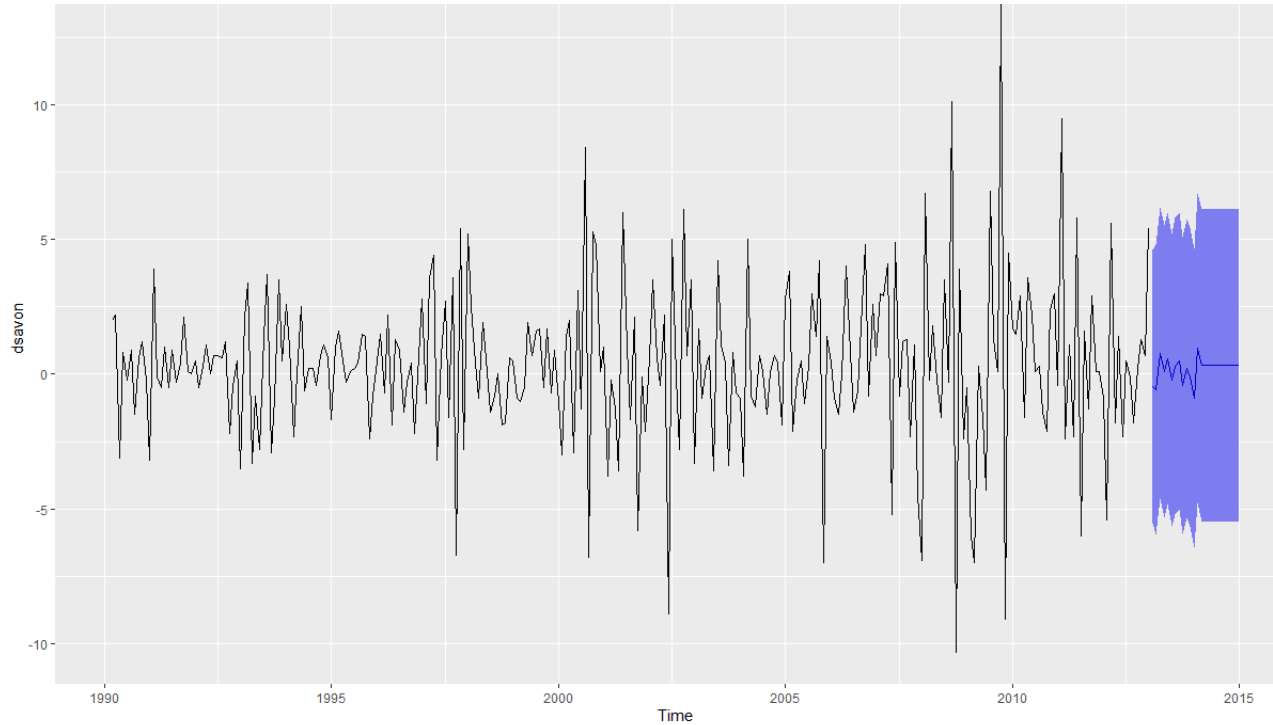


Figure 15: Estimation du modèle ARMA(0,13) à 95%

```

      lag      pval
[1,] 1 0.5520056
[2,] 2 0.8375583
[3,] 3 0.9280070
[4,] 4 0.9645807
[5,] 5 0.9403960
[6,] 6 0.9488748
[7,] 7 0.9650259
[8,] 8 0.9834692
[9,] 9 0.9885008
[10,] 10 0.9944529
[11,] 11 0.9845775
[12,] 12 0.9919378
[13,] 13 0.9959379
[14,] 14 0.9738427
[15,] 15 0.9844005
[16,] 16 0.9799516
[17,] 17 0.8482344
[18,] 18 0.8700731
[19,] 19 0.8432629
[20,] 20 0.8060293
[21,] 21 0.6944030
[22,] 22 0.6625450
[23,] 23 0.6887045
[24,] 24 0.5198339

```

Figure 16: Test du modèle ARIMA(0,1,13)

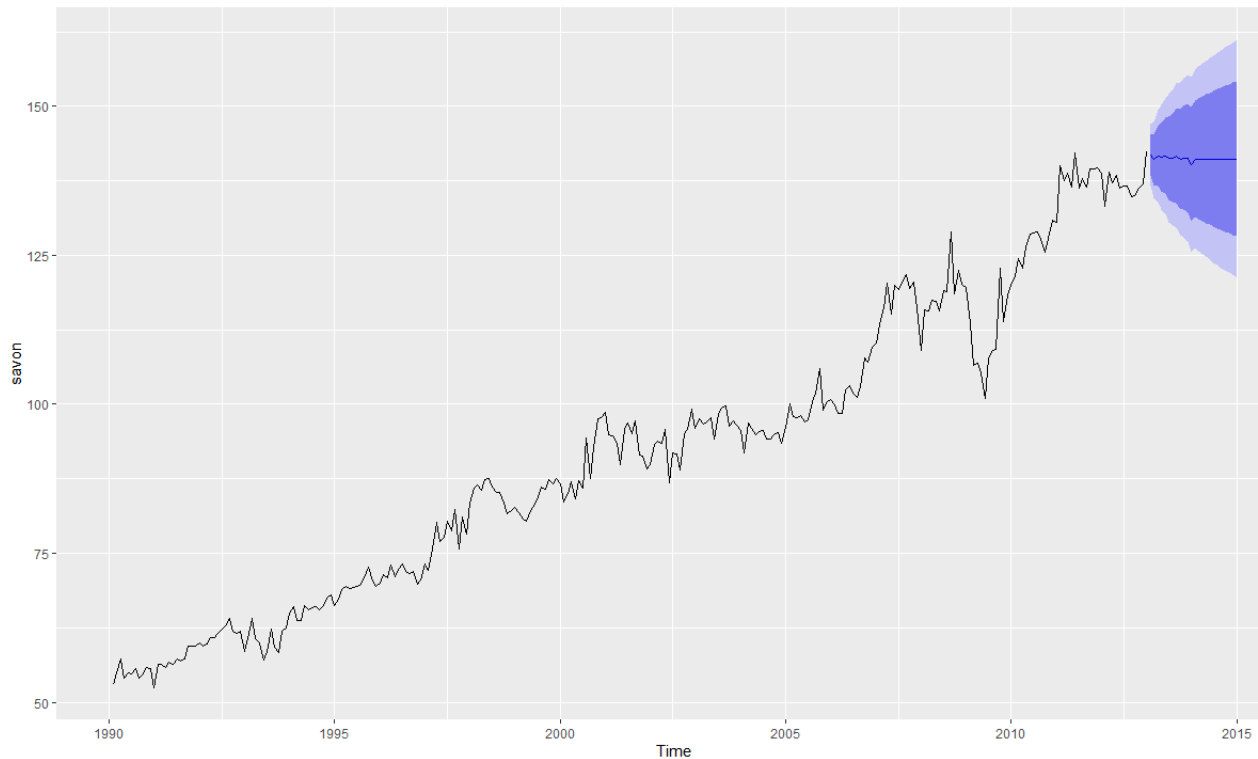


Figure 17: Estimation du modèle ARIMA(0,1,13) à 95%

4.1 Code

```
#On importe les packages necessaires
```

```
install.packages(" zoo")
```

```
install.packages(" tseries")
```

```
library(zoo)
```

```
library(tseries)
```

```
require(zoo)
```

```
require(tseries)
```

```
library(fUnitRoots)
```

```
#On importe les donnees
```

```
#lien des donnees = " https://www.insee.fr/fr/statistiques/serie/001562489#Tableau "
```

```
data <- read.csv("/Users/Rayane/Documents/valeurs_mensuelles.csv",sep=";")
```

```
View(data)
```

```
#On prepare la base de donnees afin de travailler dessus
```

```
dates_char <- as.character(data$dates)
```

```
dates_char[1]; dates_char[length(dates_char)]
```

```
dates<-as.yearmon(seq(from=1990+1/12,to=2012+12/12,by=1/12))
```

```
savon <- zoo(data$savon , order.by=dates)
```

```
plot(savon)
```

```
#Graphiquement elle ne semble pas stationnaire. Elle semble comporter une tendance lineaire
```

```
#On verifie ca avec une regression lineaire de savon sur le temps
```



```

lt<-lm(savon~dates)

r<-lt$residuals
summary(lt)

#On differencie une fois pour essayer de rendre notre STL stationnaire
dsavon <- diff(savon,1)
plot(cbind(savon,dsavon))

#On veut verifier la stationnarite, donc on effectue differents tests

#On realise tout d'abord un pp-test

pp.test(dsavon)

#On realise ensuite un ADF test, nc car pas de constante ni de trend

adf<-adfTest(dsavon, lag=1,type="nc")
adf

#On verifie que les residus du modele d'autoregressions sont bien non autocorreles
Qtests <- function(series, k, fitdf=0){
  pvals <- apply(matrix(1:k), 1, FUN=function(l){
    pval <- if (l<=fitdf) NA else Box.test(series, lag=l, type="Ljung-Box", fitdf=fitdf)$p
    return(c("lag"=l,"pval"=pval))})
  return(t(pvals))}

t(Qtests(adf@test$lm$residuals[1:24],24))

#On realise un kpss-test
kpss.test(dsavon)

#On affiche les autocorrelations et les autocorrelations
par(mfrow=c(1,2))
acf(dsavon);pacf(dsavon)

#On calibre notre modele ARMA

#affichage du test de significativite des coefficients
signif <- function(estim){
  coef <- estim$coef
  se <- sqrt(diag(estim$var.coef))
  t <- coef/se
  pval <- (1-pnorm(abs(t)))*2
  return(rbind(coef,se,pval))
}

#On estime des modeles arima et on en verifie l'ajustement et la validite

modelchoice <- function(p,q,data=dsavon, k=24){
  estim <- try(arima(data, c(p,0,q),optim.control=list(maxit=20000)))
  if (class(estim)=="try-error") return(c("p"=p,"q"=q,"arsignif"=NA,"masignif"=NA,"resnocor"=NA))
  arsignif <- if (p==0) NA else signif(estim)[3,p]<=0.05
  masignif <- if (q==0) NA else signif(estim)[3,p+q]<=0.05

```

```

resnocorr <- sum(Qtests(estim$residuals,24,length(estim$coef)-1)[,2]<=0.05,na.rm=T)==0)
checks <- c(arsignif,masignif,resnocorr)
print(checks)
ok <- as.numeric(sum(checks,na.rm=T)==(3-sum(is.na(checks))))
return(c("p"=p,"q"=q,"arsignif"=arsignif,"masignif"=masignif,"resnocorr"=resnocorr,"ok"=ok))
}

#On teste tout les arma pour trouver les bonnes valeurs de p et q
armamodelchoice <- function(pmax,qmax){
  pqs <- expand.grid(0:pmax,0:qmax)
  t(apply(matrix(1:dim(pqs)[1]),1,function(row) {
    p <- pqs[row,1]; q <- pqs[row,2]
    cat(paste0("Computing ARMA(",p," ",q," ") \n"))
    modelchoice(p,q)
  })))
}
#On a fix pmax et qmax gr ce l'acf et le pacf
pmax=11; qmax=13
T <- length(dsavon)

# On regarde les modeles bien ajustes et valides
armamodels <- armamodelchoice(pmax,qmax) #estime tous les arima
selec <- armamodels[armamodels[, "ok"]==1&!is.na(armamodels[, "ok"]),] #mod les bien ajust
selec

#On regarde tout les p et q candidats pour le mod le.
pqs <- apply(selec,1,function(row) list("p"=as.numeric(row[1]),"q"=as.numeric(row[2])))
names(pqs) <- paste0("arma(",selec[,1],",",selec[,2],")")
models <- lapply(pqs, function(pq) arima(r,c(pq[["p"]],0,pq[["q"]])) #cr e une liste des
vapply(models, FUN.VALUE=numeric(2), function(m) c("AIC"=AIC(m),"BIC"=BIC(m)))

#On determine le modele que l'on conservera en minimisant le R2 de nos modeles
adj_r2 <- function(model){
  ssres <- sum(model$residuals^2) #somme des r sidus au carr
  p <- model$arma[1]
  q <- model$arma[2]
  sstot <- sum(dsavon[-c(1:max(p,q))]^2) #somme des observations de l echantillon au c
  n <- model$nobs-max(p,q) #taille de l chantillon
  adj_r2 <- 1-(ssres/(n-p-q-1))/(sstot/(n-1)) #r2 ajust
  return(adj_r2)}
adj_r2(arma603)
adj_r2(arma0013)

#Fonction pour implementer les modeles arima
arimafit <- function(estim){
  adjust <- round(signif(estim),3)
  pvals <- Qtests(estim$residuals,36,length(estim$coef)-1)
  pvals <- matrix(apply(matrix(1:36,nrow=6),2,function(c) round(pvals[c,],3)),nrow=6)
  colnames(pvals) <- rep(c("lag", "pval"),6)
  cat("tests de nullit des coefficients :\n")
  print(adjust)
  cat("\n tests d absence d autocorrelation des r sidus : \n")
  print(pvals)
}

#On implemente notre modele arima

arma603 <- arima(dsavon,c(6,0,3)); arimafit(arma603)

```

```
arma0013<- arima(dsavon,c(0,0,13)); arimafit(arma0013)
```

```
arma603  
arma0013
```

```
#On trace la representation de la prevision      95%  
require(forecast)  
autoplot(forecast(arma603, level = c(95)))  
autoplot(forecast(arma0013, level=c(95)))
```

```
arma613 <- arima(savon,c(6,1,3),method='ML')  
arma0113 <-arima(savon,c(0,1,13),method='ML')
```

```
#On teste l'absence d'autocorrelation des residus.  
t(Qtests(arma613$residuals,24))  
Qtests(arma0113$residuals,24)
```

```
autoplot(forecast(arma0113), level=c(95))
```

```
arimafit(arma613)
```