Prédisez la demande en électricité

Pinos Bruno

SOMMAIRE

Exucative Summary

ENERCOOP souhaite un modèle permettant de déterminer la demande en électricité des utilisateurs.

Trouver un modèle efficace pour faire une telle prédiction ?

Insight et recommandation:

La modèle SARIMA(0,0,0)(0,1,1)[12] sur la consommation ajusté de la température est très fiable et répond à la problématique.

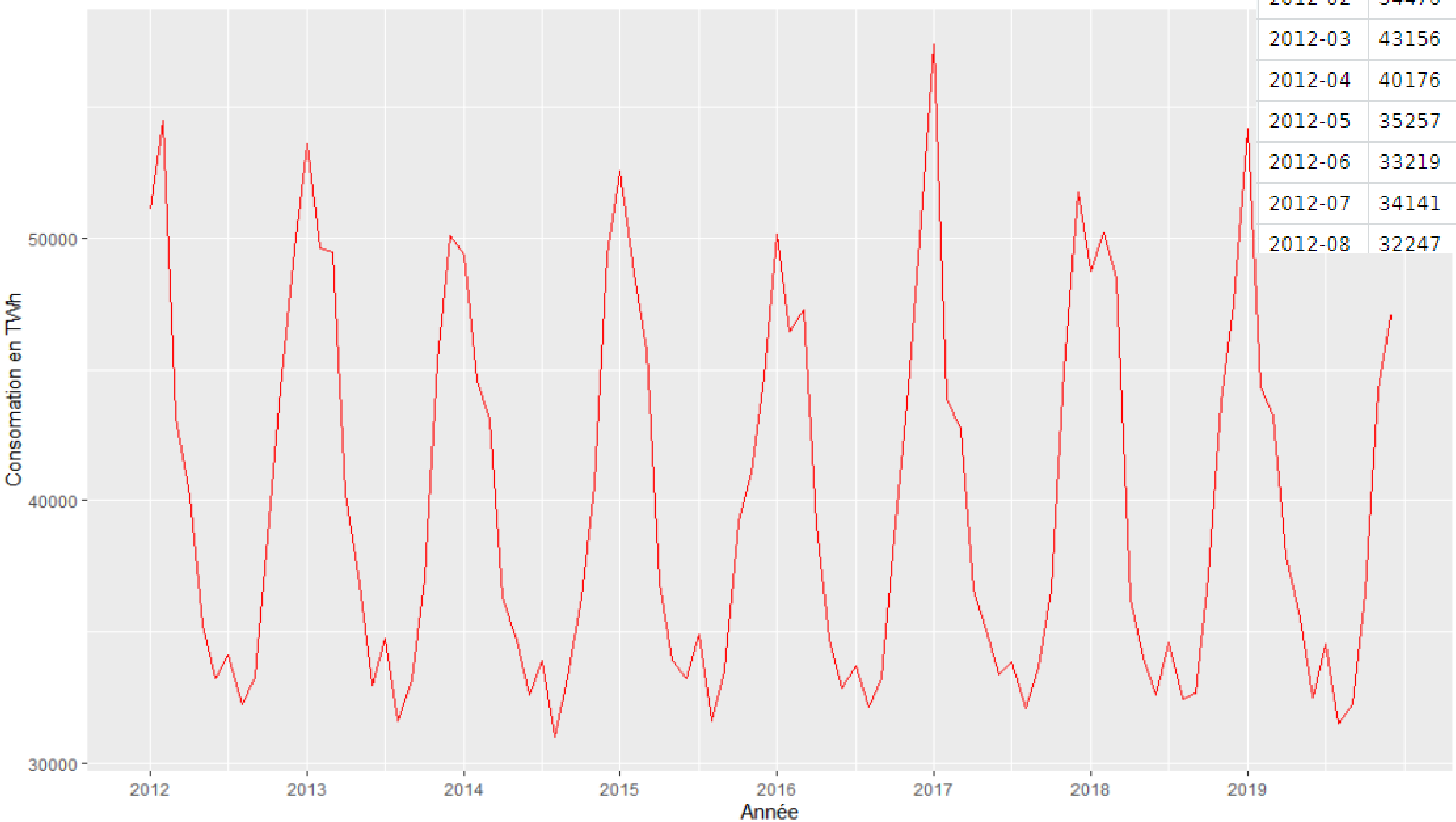
Présentation des données utilisées

« Traitement, nettoyage, analyse »

Consommation de l'électricité en France

Source : https://www.rte-france.com/eco2mix/telecharger-les-indicateurs

Date	Consommation
2012-01	51086
2012-02	54476
2012-03	43156
2012-04	40176
2012-05	35257
2012-06	33219
2012-07	34141
2012-08	32247



Degrés jours Unifiés

Source : https://cegibat.grdf.fr/simulateur/calcul-dju

Sur une période et un lieu donnés : somme des écarts entre une température de référence (généralement 18 degrés) et la température moyenne journalière.

Données manquantes en 2020

Année	JAN	FÉV	MAR	AVR	MAI	JUN	JUI	AOÛ	SEP	OCT	NOV	DÉC
2020	339.0	249.6	269.2	105.6	85.9	36.2	0.0	0.0	0.0	0.0	0.0	0.0
2019	404.9	268.8	234.3	177.2	126.8	35.7	10.0	11.8	45.3	136.9	282.6	327.3
2018	303.4	432.6	314.4	128.9	74.1	20.1	1.6	13.5	53.7	133.4	282.8	325.9
2017	467.9	278.4	208.9	187.1	87.8	21.4	13.5	20.8	74.5	109.3	282.6	369.0
2016	364.4	321.6	321.1	214.1	100.7	40.8	14.7	13.8	27.3	177.3	285.6	390.8
2015	392.0	365.7	275.7	154.6	104.8	36.4	15.8	16.1	81.8	179.6	196.9	248.1
2014	324.4	281.9	226.6	141.8	112.5	37.3	17.7	36.1	34.4	101.3	223.3	368.2
2013	429.2	402.2	376.6	216.3	161.5	57.6	9.3	19.9	55.3	112.0	303.9	349.5
2012	336.0	435.9	210.5	231.2	93.8	50.4	29.8	14.1	72.6	159.2	296.2	345.9

Degrés jours Unifiés

```r
#Je récupère les données DJU de paris(~France)
DJU<-read.csv("DJU.csv", sep = ",", encoding = "latin1", skip = 11)

#Je rajoute les données DJU à ma table principale
for (j in 0:7) {
for (i in 1:12) {
  conso$DJU[i+j*12]<-DJU[9-j,i+1]
}
}
```

	Date	Territoire	Qualité	Conso	DJU
1	2012-01	France	Données définitives	51086	336.0
2	2012-02	France	Données définitives	54476	435.9
3	2012-03	France	Données définitives	43156	210.5
4	2012-04	France	Données définitives	40176	231.2
5	2012-05	France	Données définitives	35257	93.8
6	2012-06	France	Données définitives	33219	50.4
7	2012-07	France	Données définitives	34141	29.8

Colonne DJU remplit par les données téléchargées

Correction de l’effet température

« modèle, indicatrice mois, time »

Modèle

$$y = at + bu + \sum_{i=1}^{12} c_i 1$$

t représente l'évolution temporelle
u correspond au DJU du mois en cours
c L'indicatrice du mois en cours

Création indicatrice mois

```
#Je crée une indicatrice mois dans ma table principale
for (i in 1:12){
  su=rep(0,times=12)
  su[i]=1
  s=rep(su,times=8)
  assign(paste("s",i,sep=""),s)
}

conso_lm<-cbind(conso,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12)
```

Date	Conso	DJU	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12
2012-01	51086	336.0	1	0	0	0	0	0	0	0	0	0	0	0
2012-02	54476	435.9	0	1	0	0	0	0	0	0	0	0	0	0
2012-03	43156	210.5	0	0	1	0	0	0	0	0	0	0	0	0
2012-04	40176	231.2	0	0	0	1	0	0	0	0	0	0	0	0
2012-05	35257	93.8	0	0	0	0	1	0	0	0	0	0	0	0
2012-06	33219	50.4	0	0	0	0	0	1	0	0	0	0	0	0
2012-07	34141	29.8	0	0	0	0	0	0	1	0	0	0	0	0
2012-08	32247	14.1	0	0	0	0	0	0	0	1	0	0	0	0
2012-09	33269	72.6	0	0	0	0	0	0	0	0	1	0	0	0
2012-10	38628	159.2	0	0	0	0	0	0	0	0	0	1	0	0
2012-11	44260	296.2	0	0	0	0	0	0	0	0	0	0	1	0
2012-12	49602	345.9	0	0	0	0	0	0	0	0	0	0	0	1

Création d’une colonne « time »

```
#Je crée une colonne time qui est une clé primaire et qui me permettra
#d'étudier et de représenter la série temporelle
conso$time<-1:nrow(conso)
```

Date	Conso	time	DJU	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10
2012-01	51086	1	336.0	1	0	0	0	0	0	0	0	0	0
2012-02	54476	2	435.9	0	1	0	0	0	0	0	0	0	0
2012-03	43156	3	210.5	0	0	1	0	0	0	0	0	0	0
2012-04	40176	4	231.2	0	0	0	1	0	0	0	0	0	0
2012-05	35257	5	93.8	0	0	0	0	1	0	0	0	0	0
2012-06	33219	6	50.4	0	0	0	0	0	1	0	0	0	0
2012-07	34141	7	29.8	0	0	0	0	0	0	1	0	0	0

Régression linéaire

On récupère le coefficient b de notre modèle

```
Call:
lm(formula = Conso ~ time + s1 + s2 + s3 + s4 + s5 + s6 + s7 +
    s8 + s9 + s10 + s11 + s12 - 1 + DJU, data = conso_lm)

Residuals:
    Min      1Q  Median      3Q     Max
-1626.9  -521.5  -115.2   488.1  2838.0

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
time    -1.493      2.972  -0.502    0.617
s1   35813.991    906.167  39.523   <2e-16 ***
s2   32735.317    846.452  38.674   <2e-16 ***
s3   33709.936    691.500  48.749   <2e-16 ***
s4   30151.066    523.577  57.587   <2e-16 ***
s5   30399.377    405.760  74.920   <2e-16 ***
s6   31373.101    330.086  95.045   <2e-16 ***
s7   33773.656    318.705 105.971   <2e-16 ***
s8   31112.946    321.857  96.667   <2e-16 ***
s9   30796.635    349.295  88.168   <2e-16 ***
s10  31559.118    458.817  68.784   <2e-16 ***
s11  31957.239    696.362  45.892   <2e-16 ***
s12  34166.294    840.052  40.672   <2e-16 ***
DJU     43.372      2.202  19.699   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 789.6 on 82 degrees of freedom
Multiple R-squared:  0.9997,    Adjusted R-squared:  0.9996
F-statistic: 1.817e+04 on 14 and 82 DF,  p-value: < 2.2e-16
```

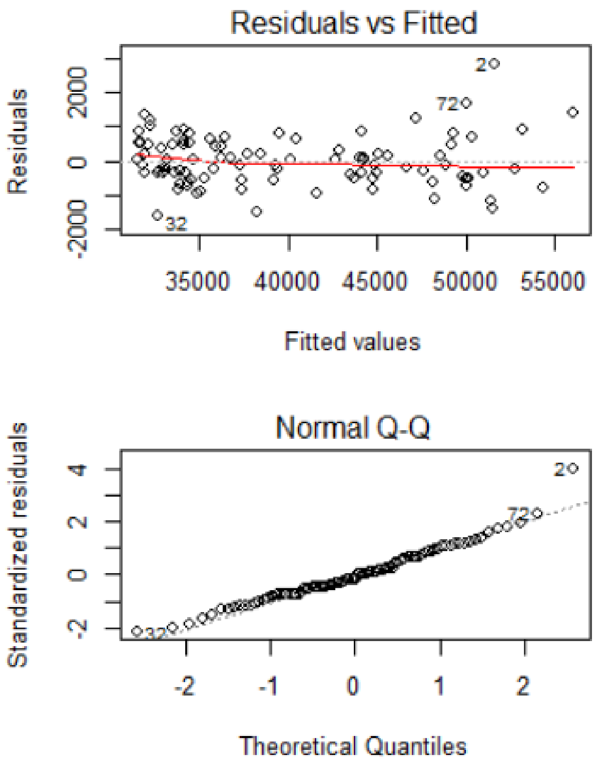
Tests de validités

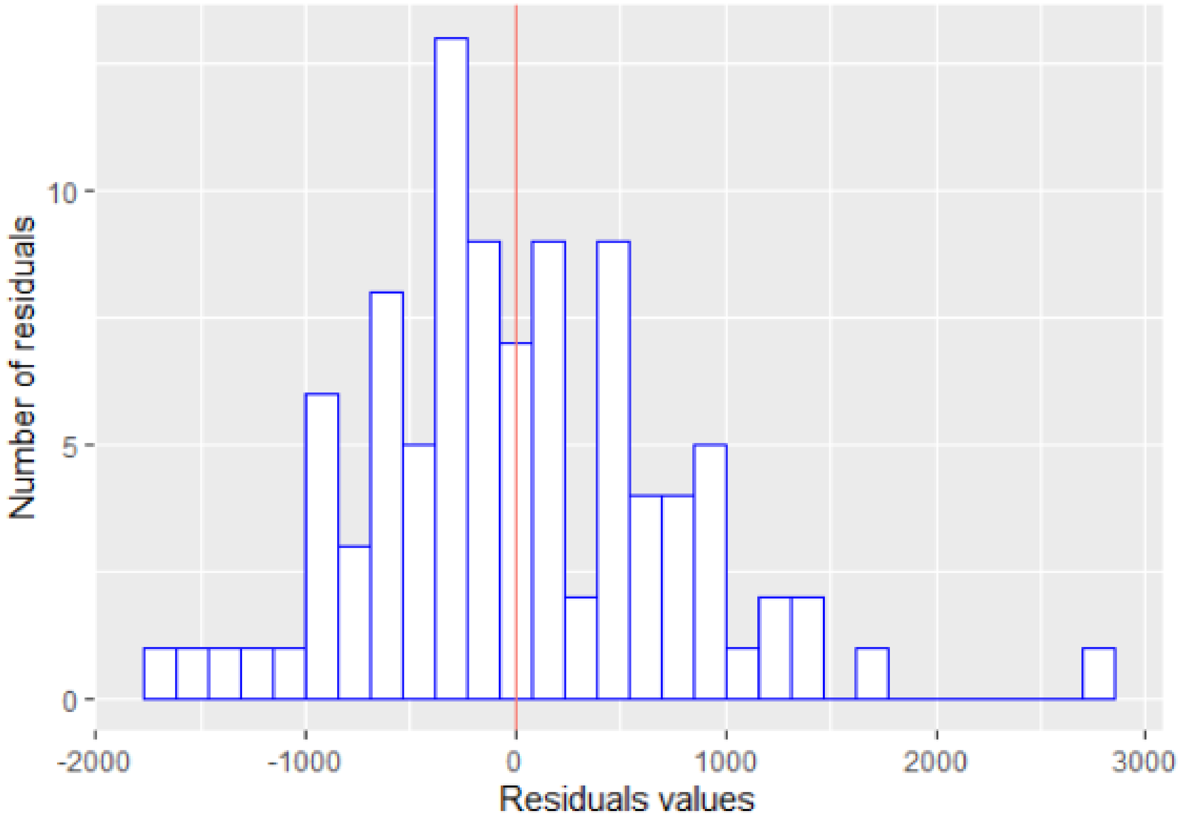
```
        studentized Breusch-Pagan test

data:  fit
BP = 16.915, df = 13, p-value = 0.2032
```

```
        Shapiro-Wilk normality test

data:  fit$residuals
W = 0.97464, p-value = 0.05936
```





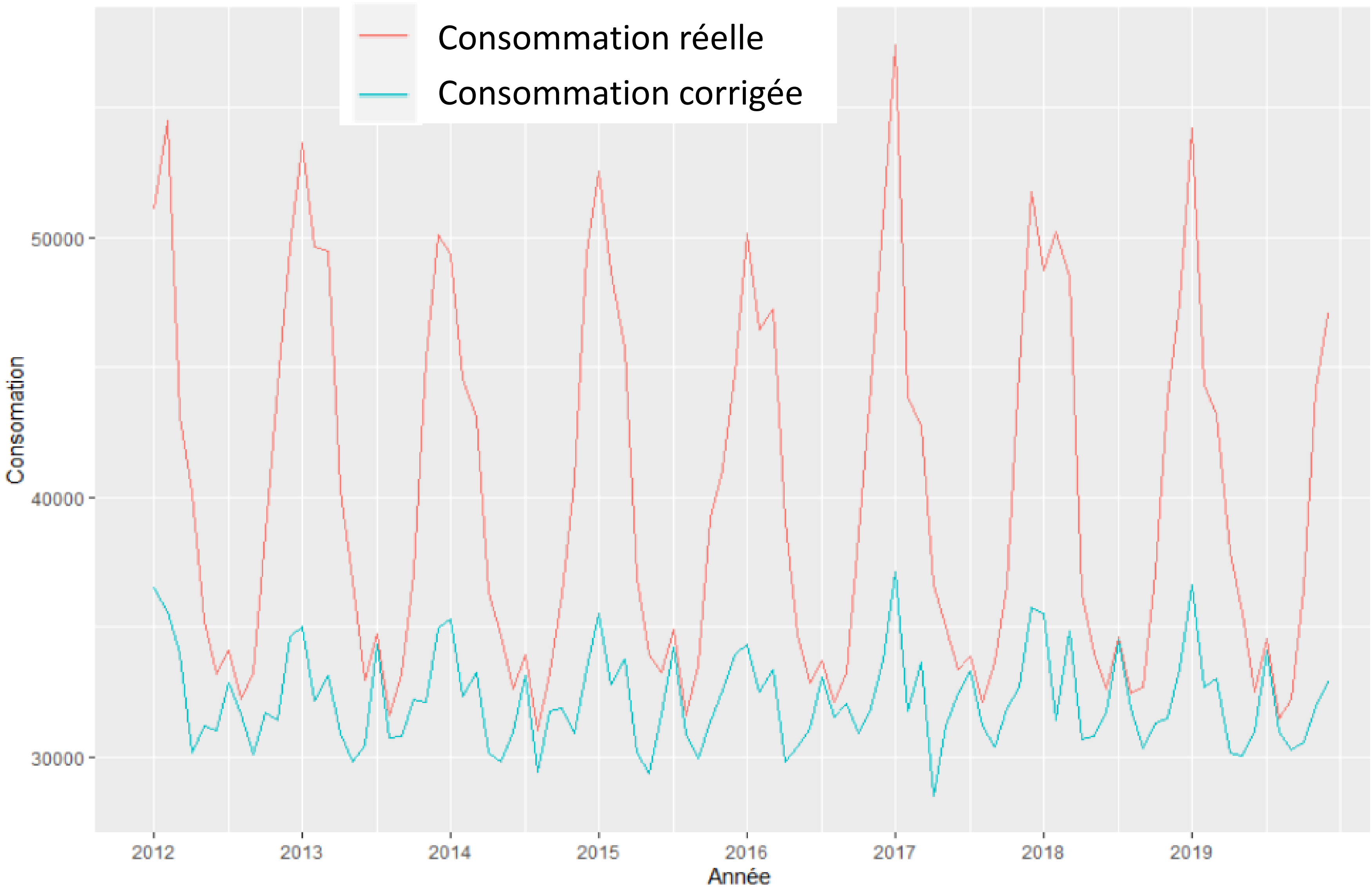
Modèle corrigé

$$y_{corrigé} = y - bu = at + \sum_{i=1}^{12} C_i 1$$

```
#J'ajoute une colonne qui donne la consomation corrigée de l'effet
#température
conso$Conso_adj<-conso$Conso - conso$DJU*fit$coefficients[["DJU"]]
```

Date	Territoire	Qualité	Conso	time	DJU	Conso_adj
2012-01	France	Données définitives	51086	1	336.0	36513.14
2012-02	France	Données définitives	54476	2	435.9	35570.32
2012-03	France	Données définitives	43156	3	210.5	34026.28
2012-04	France	Données définitives	40176	4	231.2	30148.49
2012-05	France	Données définitives	35257	5	93.8	31188.74
2012-06	France	Données définitives	33219	6	50.4	31033.07

Consommation corrigée de l'effet température vs consommation réelle

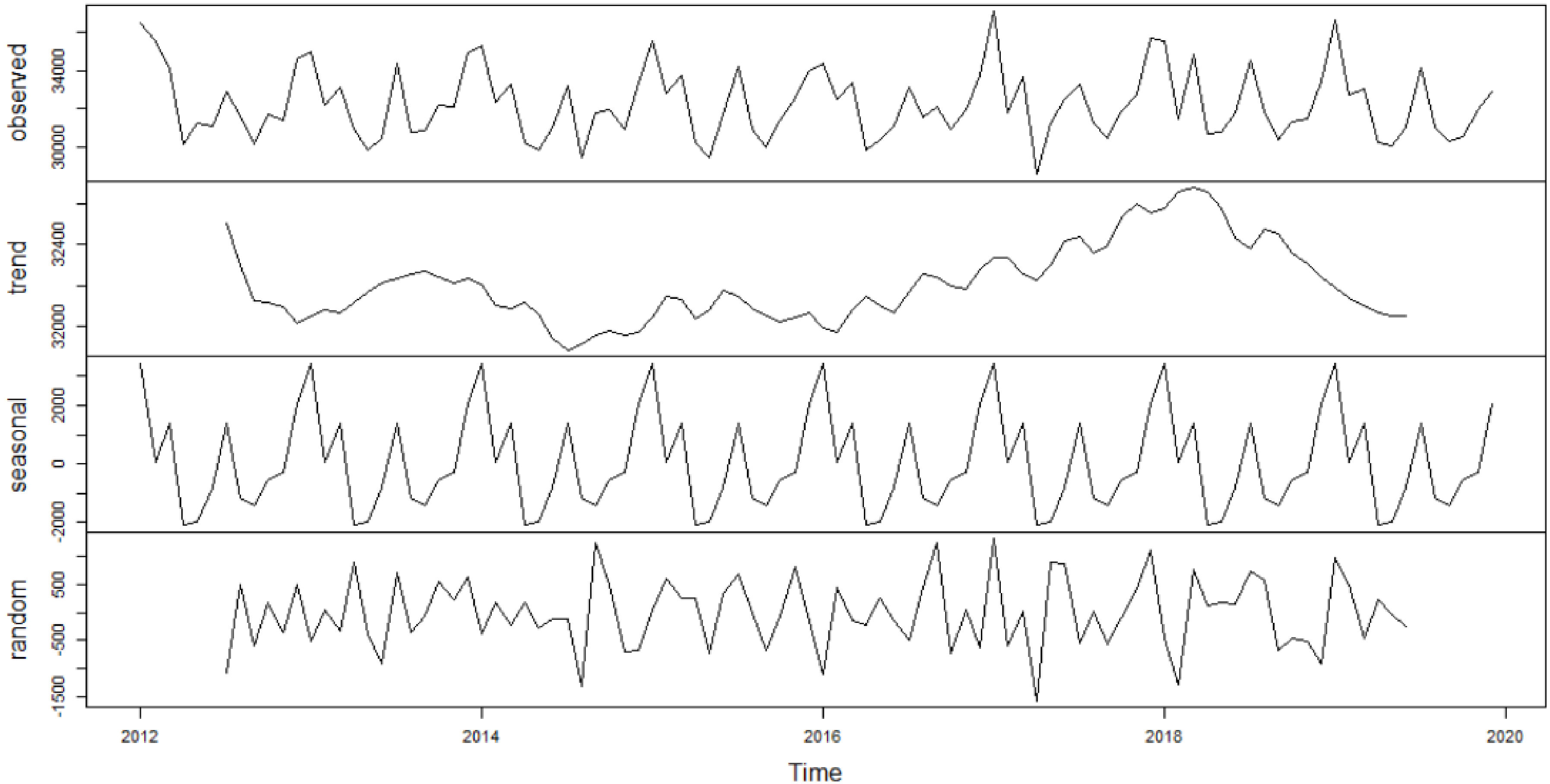


Dessaisonalisation

« grâce au moyenne mobile »

Décomposition

```
#J'utilise la fonction decompose pour étudier ma série temporelle
decomp_conso_adj<-decompose(ts_conso_adj)
plot(decomp_conso_adj)
```

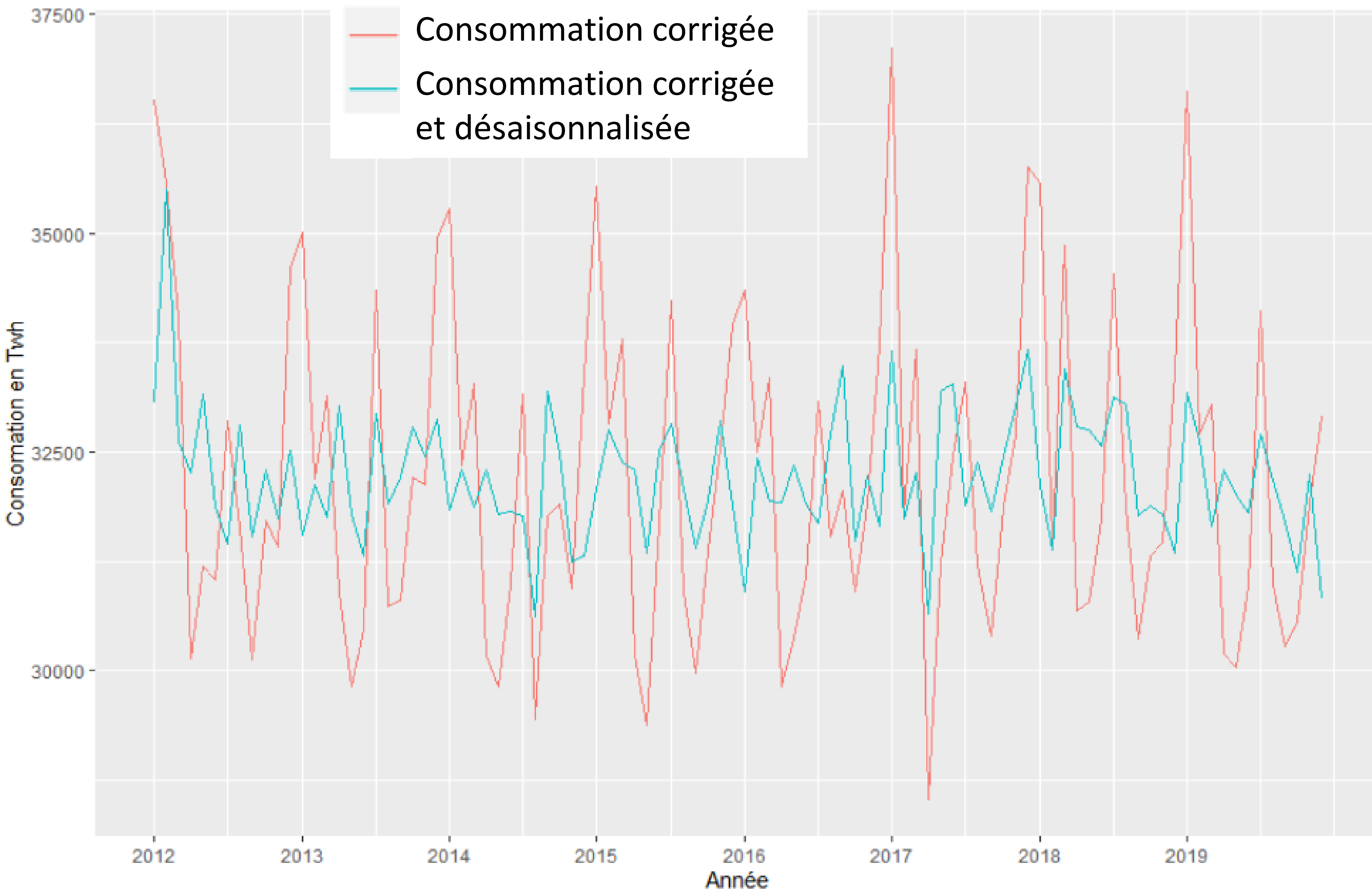


Dessaisonalisation

```
#Je crée une colonne avec la consommation dessaisonalisée
conso$deseasonal<-conso$Conso_adj - conso$seasonal
```

Date	Territoire	Qualité	Conso	time	DJU	Conso_adj	deseasonal
2012-01	France	Données définitives	51086	1	336.0	36513.14	33059.26
2012-02	France	Données définitives	54476	2	435.9	35570.32	35505.62
2012-03	France	Données définitives	43156	3	210.5	34026.28	32626.01
2012-04	France	Données définitives	40176	4	231.2	30148.49	32257.18
2012-05	France	Données définitives	35257	5	93.8	31188.74	33159.99
2012-06	France	Données définitives	33219	6	50.4	31033.07	31863.88

Consommation corrigée vs consommation corrigée et désaisonnalisée

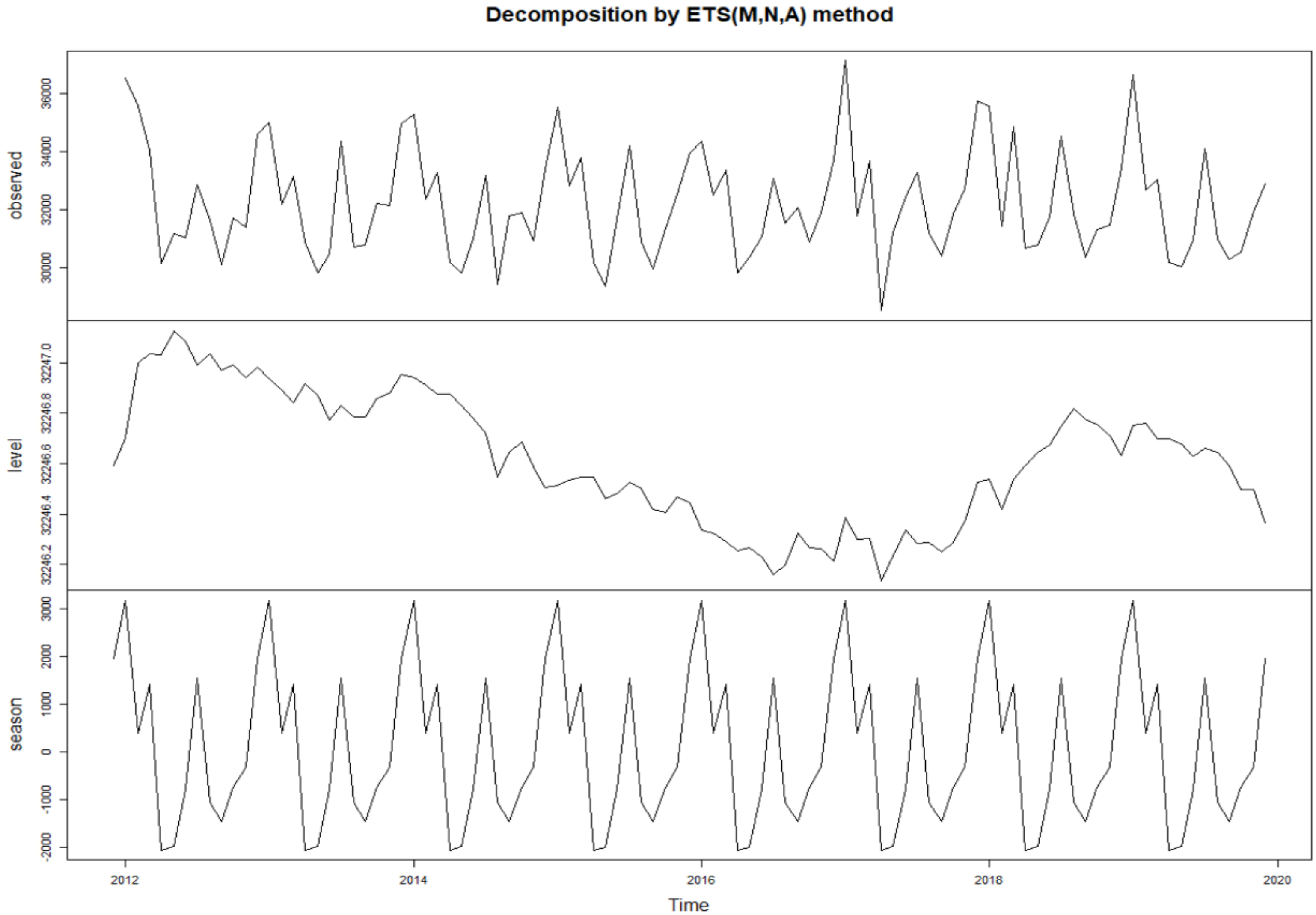


Prévision de la consommation méthode Holt-Winters

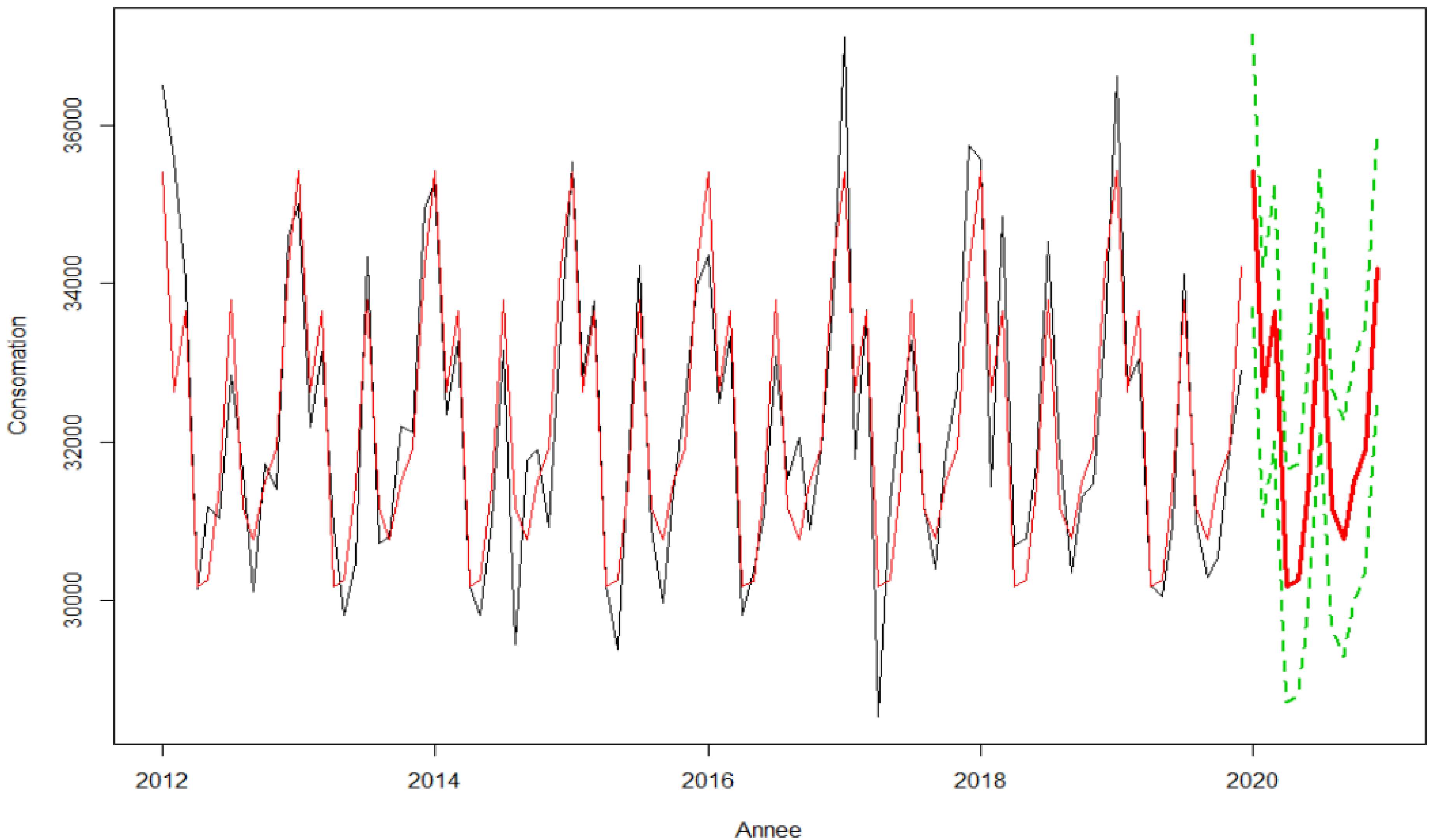
« résultats, analyses »

Décomposition

```
#J'utilise le lissage exponentiel Holt Winter pour effectuer des
#prédictions
hw<-ets(ts_conso_adj, model = "ZZZ")
```



Courbe de prédiction



Analyse du modèle

```
ETS(M,N,A)

Call:
 ets(y = ts_conso_adj, model = "ZZZ")

  Smoothing parameters:
    alpha = 1e-04
    gamma = 0.0023

  Initial states:
    l = 32246.5919
    s = 1960.557 -331.9638 -741.9747 -1471.627 -1074.929 1549.063
           -787.3409 -1994.793 -2065.057 1402.805 386.6649 3168.596

  sigma:  0.0248

     AIC     AICc      BIC
1735.966 1741.966 1774.432

Training set error measures:
                    ME     RMSE      MAE        MPE     MAPE      MASE
Training set -23.32789 742.9247 574.2253 -0.1325487 1.771785 0.6437558
                   ACF1
Training set 0.02388857
```

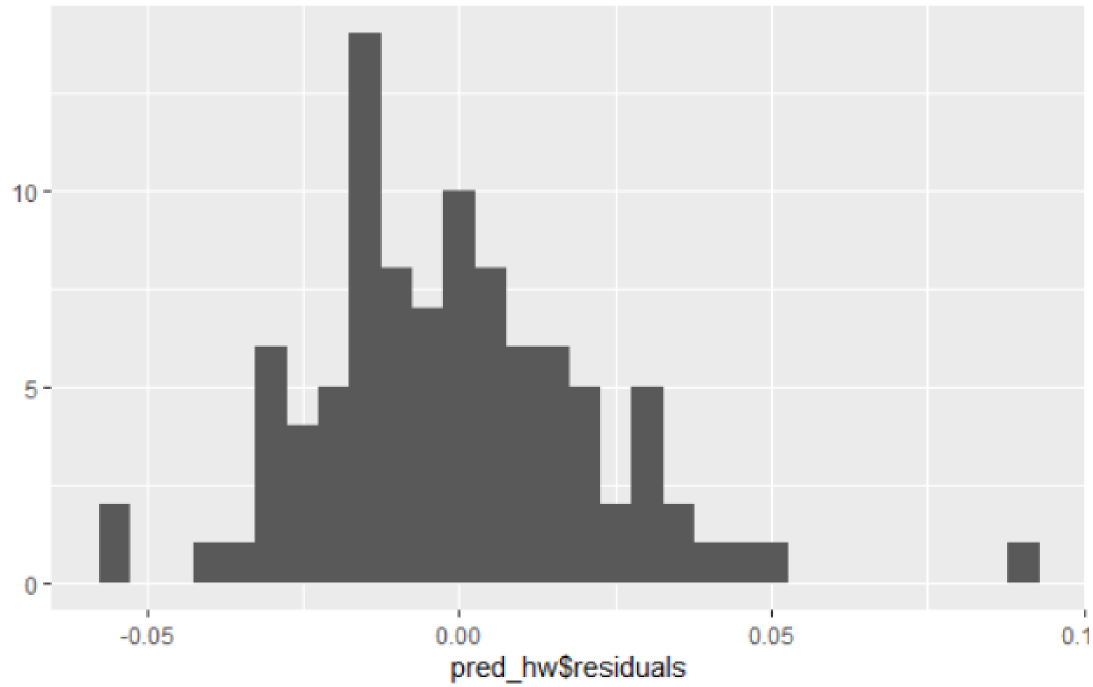
Tests de validités

Test Q de Ljung-Box :

Retard	p-value
6	0.76714
12	0.62892
18	0.33000
24	0.41346
30	0.32081
36	0.31231

```
Shapiro-Wilk normality test

data:  pred_hw$residuals
W = 0.96965, p-value = 0.02514
```

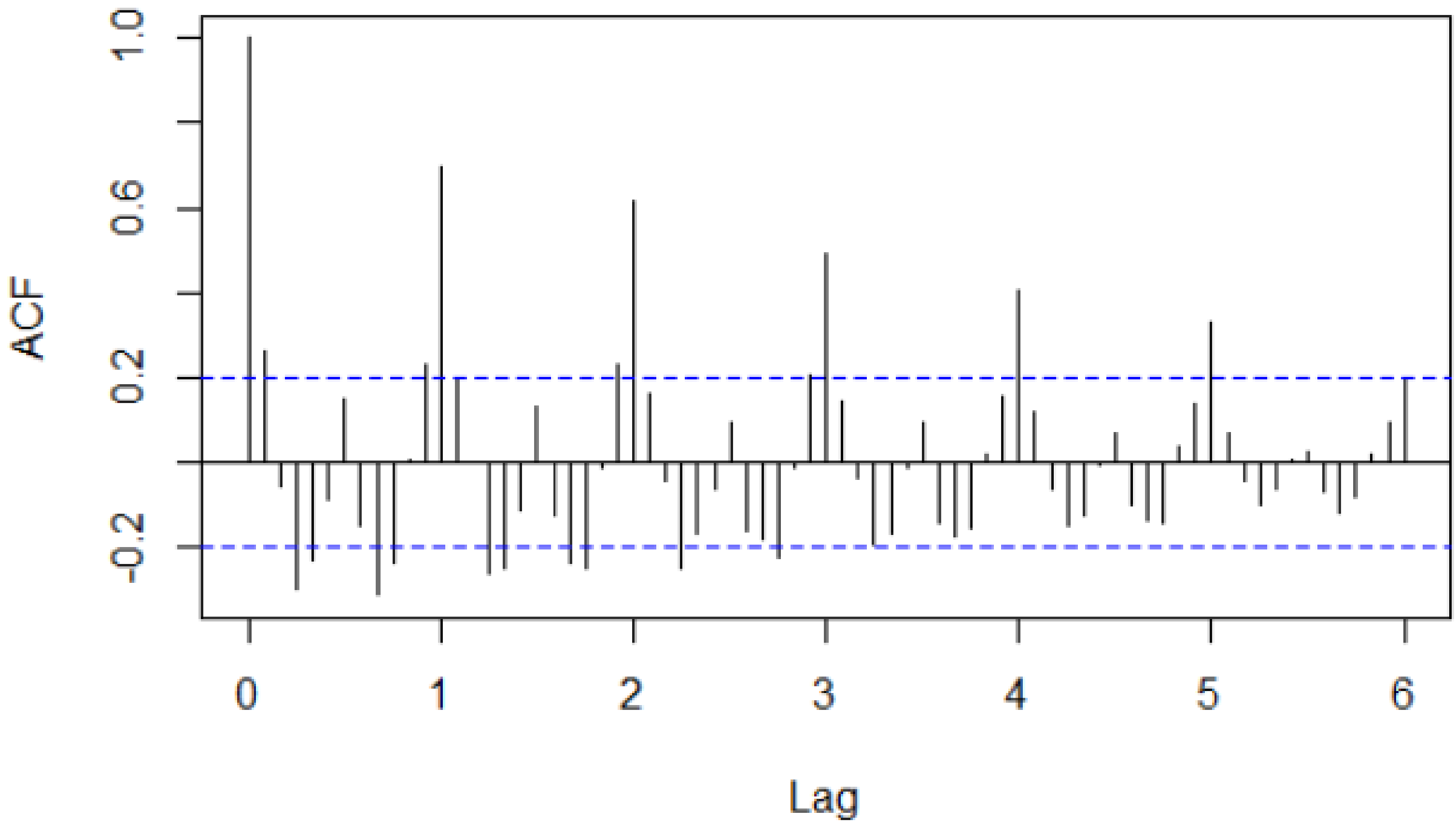


Prévision de la consommation méthode SARIMA

« résultats, tests, analyses »

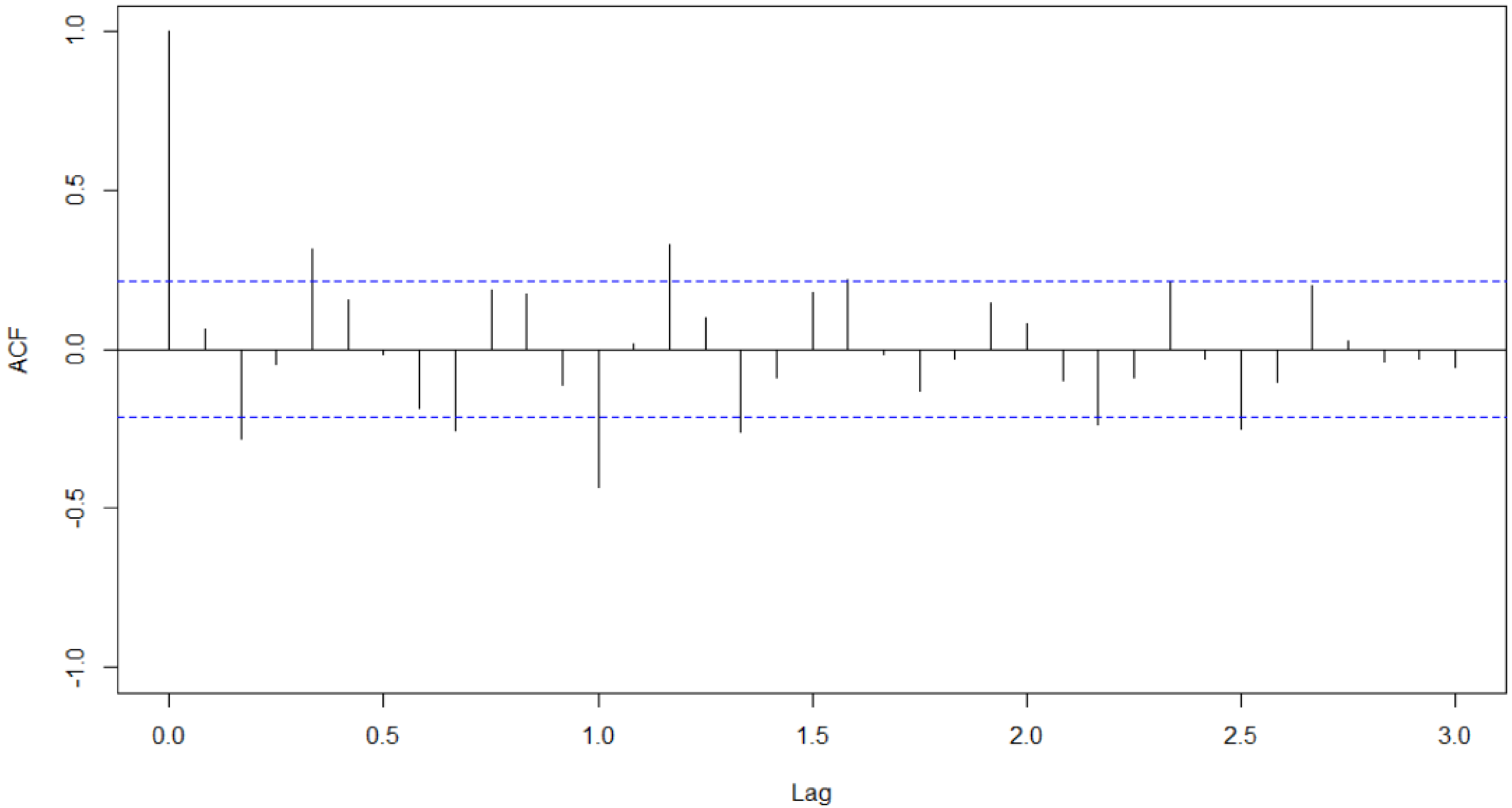
(I)

```
#J'étudie l'autocorrelogramme de ma série pour étudier comment la série
#doit être différenciée
acf(ts_conso_adj,lag.max=72, plot=TRUE)
```

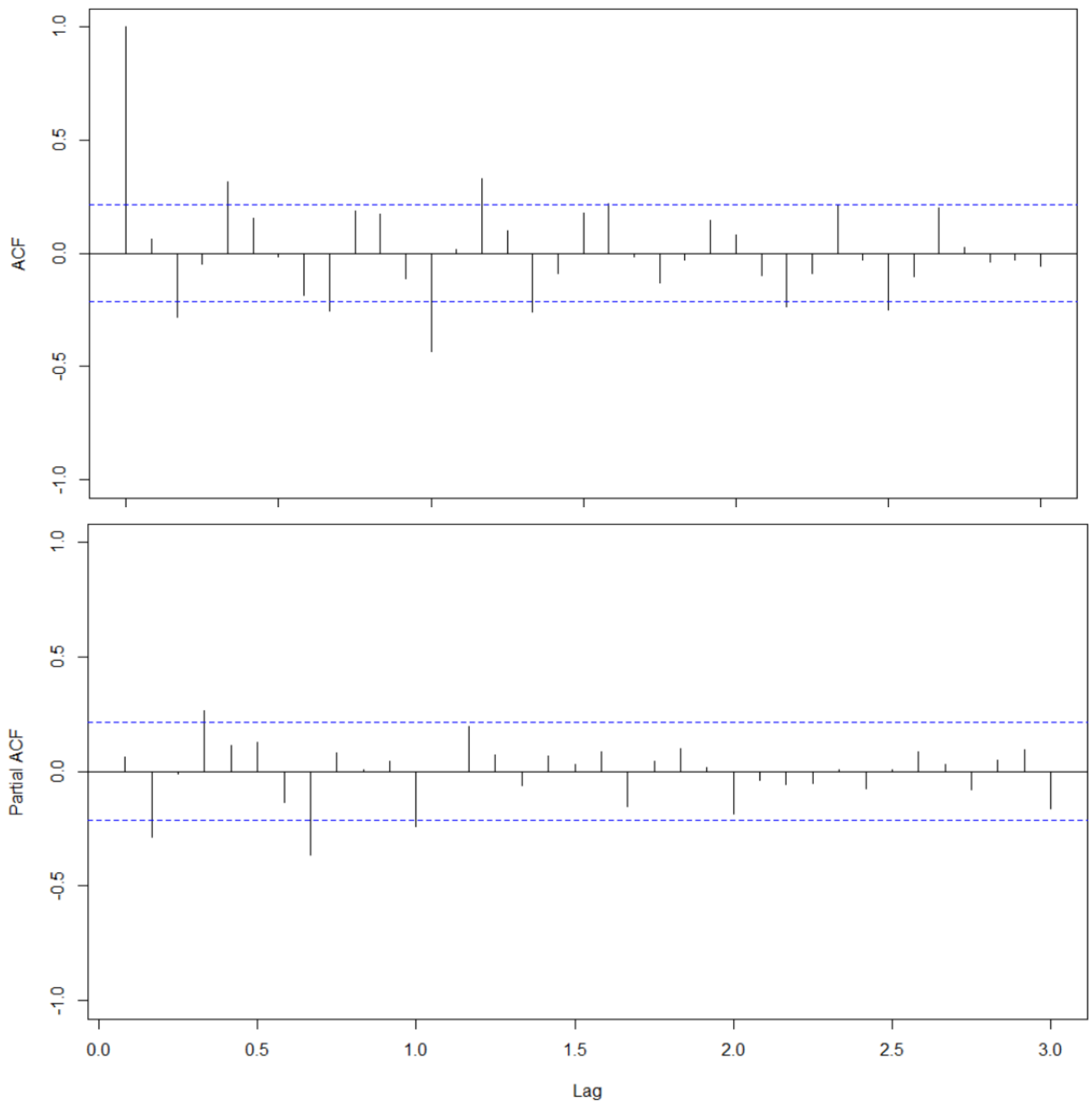


(I – B12)

```
#J'effectue une diférenciation de lag = 12
y_dif_12=diff(ts_conso_adj,lag=12,differences=1)
```



Partial ACF



Classement des différents modèles

```
pmax = 4
qmax = 4
Pmax = 0
Qmax = 1
i<-1
c<-data.frame()

for (p in 0:pmax) {
  for (q in 0:qmax) {
    for (p12 in 0:Pmax) {
      for (q12 in 0:Qmax) {
        model1=try(Arima(ts_conso_adj,order=c(p,0,q),
             list(order=c(p12,1,q12),period=12),
             include.mean=TRUE,method="CSS-ML"),TRUE)
        if (class(try(length(t_stat(model1)),TRUE)) == "integer"){
        if (length(t_stat(model1)) == 0){
        c[i,"nom_model"]<-paste("model(",p,"0",q,")(",p12,"1",q12,")",
                                    sep = "")
        c[i,"aic"]<-model1$aic
        c[i,"bic"]<-model1$bic
        c[i,"RMCE"]<-(model1$residuals)^2%>%mean()%>%.^0.5
        i<-i+1
        }
        else if (length(t_stat(model1)[2,]) ==
          try((t_stat(model1)[2,]<0.05)%>%as.numeric()%>%sum())) {
        c[i,"nom_model"]<-paste("model(",p,"0",q,")(",p12,"1",q12,")",
                                    sep = "")
        c[i,"aic"]<-model1$aic
        c[i,"bic"]<-model1$bic
        c[i,"RMCE"]<-(model1$residuals)^2%>%mean()%>%.^0.5
        i<-i+1
        }
        }
      }
    }
  }
}
```

nom_model	aic	bic	RMCE
model(203)(011)	1383.764	1400.780	731.1075
model(202)(011)	1386.271	1400.856	741.5489
model(000)(011)	1386.326	1391.188	772.1475
model(101)(011)	1390.152	1399.875	774.0849
model(202)(010)	1404.275	1416.430	881.6530
model(203)(010)	1406.973	1421.558	885.8001
model(000)(010)	1415.787	1418.217	1022.2439

Modèle choisi

```
Series: ts_conso_adj
ARIMA(0,0,0)(0,1,1)[12]

Coefficients:
         sma1
      -0.8714
s.e.   0.2563

sigma^2 estimated as 689594:  log likelihood=-691.16
AIC=1386.33   AICc=1386.47   BIC=1391.19

Training set error measures:
                    ME     RMSE      MAE        MPE     MAPE      MASE
Training set -82.77066 772.1475 596.4574 -0.2974778 1.852959 0.6686799
                   ACF1
Training set 0.01903371
```

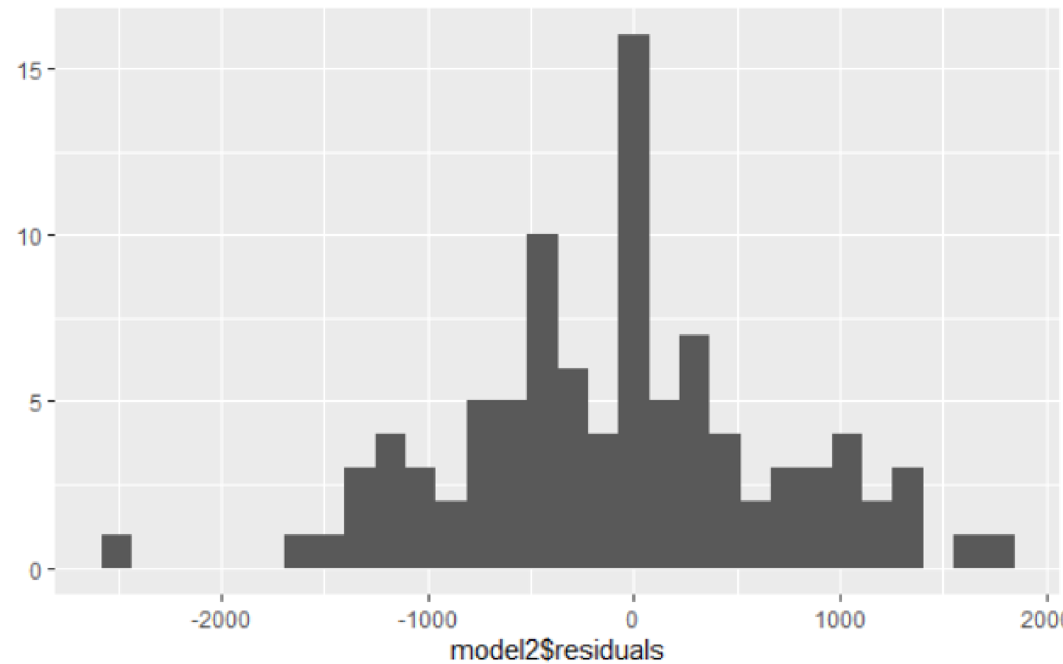
Tests de validités

Test Q de Ljung-Box :

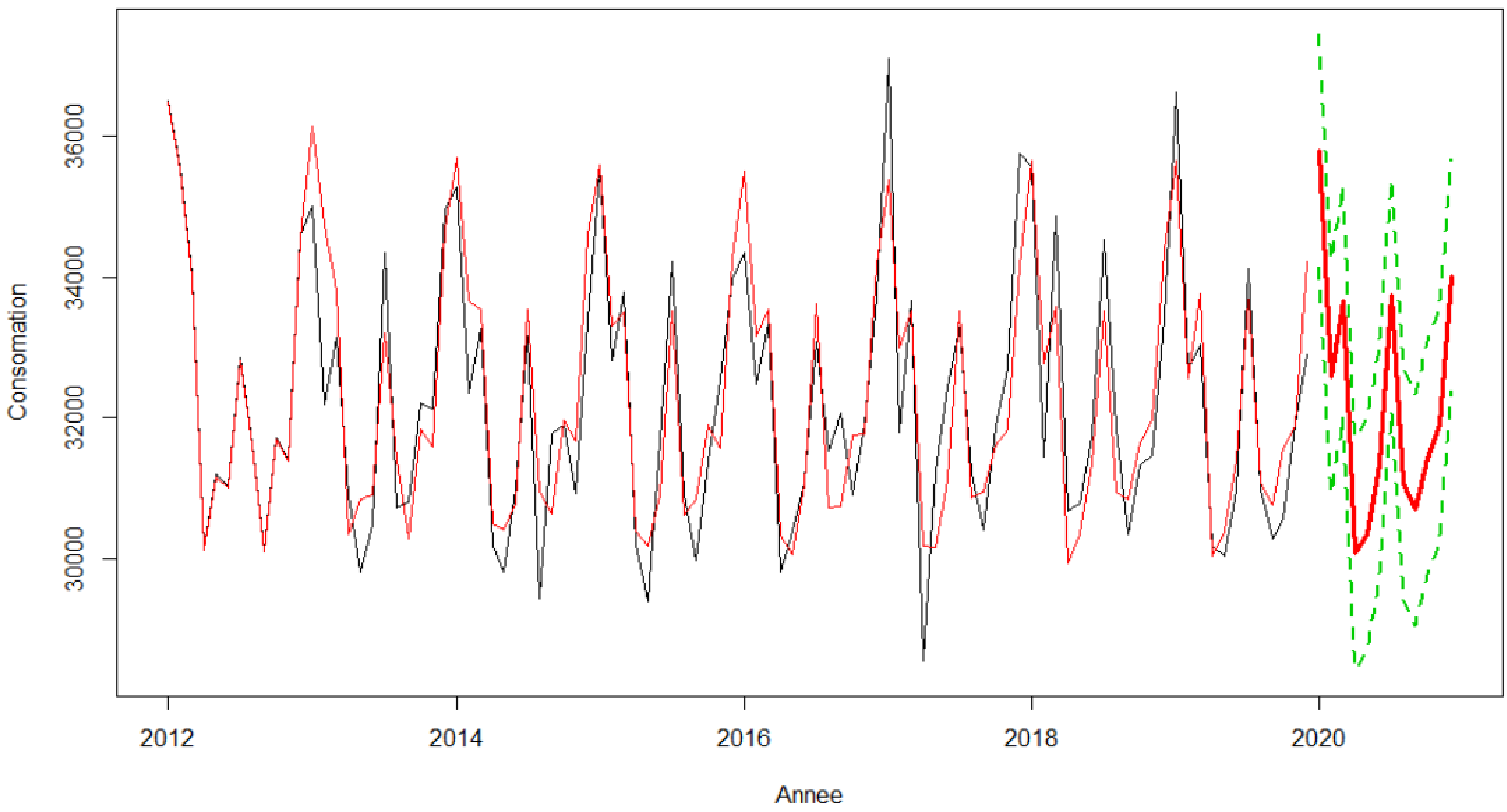
Retard	p-value
6	0.77869
12	0.49150
18	0.22727
24	0.34362
30	0.32006
36	0.39712

```
Shapiro-Wilk normality test

data:  model2$residuals
W = 0.98952, p-value = 0.6535
```



Courbe de prédiction



Auto SARIMA

```
#J'utilise une fonction qui permet de trouver des modèles performants
#de façon automatique
model2<-auto.arima(ts_conso_adj)
```

```
Series: ts_conso_adj
ARIMA(0,0,0)(0,1,1)[12]

Coefficients:
         sma1
      -0.8714
s.e.   0.2563

sigma^2 estimated as 689594:  log likelihood=-691.16
AIC=1386.33   AICc=1386.47   BIC=1391.19

Training set error measures:
                   ME     RMSE      MAE        MPE     MAPE      MASE
Training set -82.77066 772.1475 596.4574 -0.2974778 1.852959 0.6686799
                   ACF1
Training set 0.01903371
```

Conclusion

Le modèle **SARIMA(0,0,0)(0,1,1)[12]** est un peu moins précis mais me semble plus sûre, un critère d'Akaike beaucoup plus faible et une normalité des résidus très forte.

Je choisis donc le modèle **SARIMA(0,0,0)(0,1,1)[12]**

Bilan

Ce projet m’a permis d’apprendre de nombreuses choses sur les séries temporelles et leurs méthodes de prédictions.

Une étude uniquement sur une région et prenant en compte les DJU climaticien m’aurait paru plus fiable.