



prédisez une demande en electricite

ACHAT HAYAT

# Données Consommation en Électricité

```
cons_eng12= pd.read_csv("D:\\Aopenclassroom\\p9\\données/eCO2mix_RTE_Annuel-Definitif_2012.csv")
cons_eng13= pd.read_csv("D:\\Aopenclassroom\\p9\\données/eCO2mix_RTE_Annuel-Definitif_2013.csv")
cons_eng14= pd.read_csv("D:\\Aopenclassroom\\p9\\données/eCO2mix_RTE_Annuel-Definitif_2014.csv")
cons_eng15= pd.read_csv("D:\\Aopenclassroom\\p9\\données/eCO2mix_RTE_Annuel-Definitif_2015.csv")
cons_eng16= pd.read_csv("D:\\Aopenclassroom\\p9\\données/eCO2mix_RTE_Annuel-Definitif_2016.csv")
cons_eng17= pd.read_csv("D:\\Aopenclassroom\\p9\\données/eCO2mix_RTE_Annuel-Definitif_2017.csv")
cons_eng18= pd.read_csv("D:\\Aopenclassroom\\p9\\données/eCO2mix_RTE_Annuel-Definitif_2018.csv")
cons_eng19= pd.read_csv("D:\\Aopenclassroom\\p9\\données/eCO2mix_RTE_Annuel-Definitif_2019.csv")
cons_eng20= pd.read_csv("D:\\Aopenclassroom\\p9\\données/eCO2mix_RTE_Annuel-Definitif_2020.csv")
```

```
cons_eng=pd.concat([cons_eng12,cons_eng13,cons_eng14,cons_eng15,cons_eng16,cons_eng17,cons_eng18,cons_eng19,cons_eng20], ignore_index=True)
cons_eng.head()
```

	Périmètre	Nature	Date	Heures	Consommation	Prévision J-1	Prévision J	Fioul	Charbon	Gaz	...	Hydraulique - Fil de l'eau + éclusée	Hydraulique - Lacs	Hydraulique - STEP turbinage	Bioénergies - Déchets	E
0	France	Données définitives	2012-01-01	00:00	58315.0	58200.0	58200.0	492.0	25.0	3816.0	...	ND	ND	ND	ND	
1	France	Données définitives	2012-01-01	00:15	NaN	57700.0	57550.0	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	

# Données Consommation en Électricité

```
cons_eng=cons_eng.loc[cons_eng.Périmètre=='France']
```

```
nan1=cons_eng.loc[cons_eng['Consommation'].isna()==True]  
nan1
```

```
cons_eng=cons_eng[['Date','Nature','Consommation']]  
cons_eng
```

	Date	Nature	Consommation
0	2012-01-01	Données définitives	58315.0
1	2012-01-01	Données définitives	NaN
2	2012-01-01	Données définitives	58315.0

```
cons_eng['Date']=pd.to_datetime(cons_eng['Date'])  
cons_eng
```

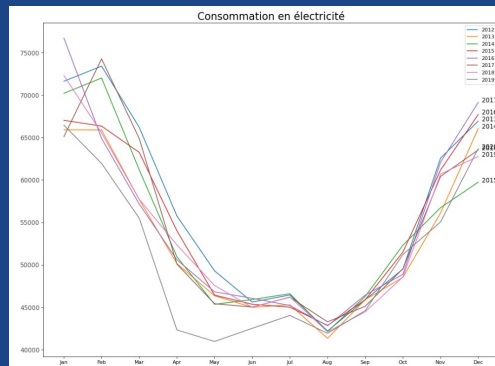
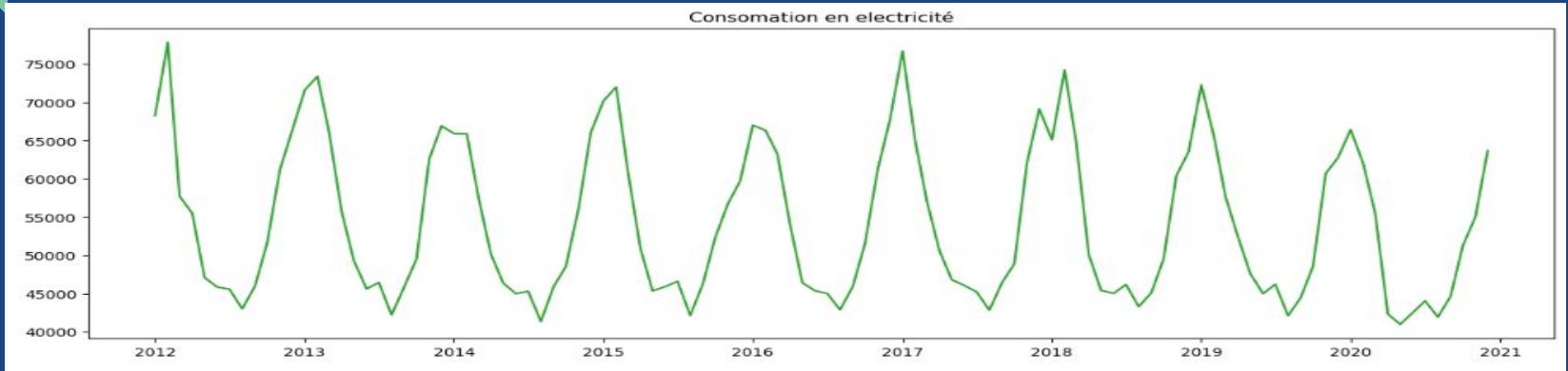
	Date	Nature	Consommation
0	2012-01-01	Données définitives	58315.0
1	2012-01-01	Données définitives	NaN

- périmètre==France
- transformer en date time ['date']
- grouper par mois les données consommation

```
cons_eng_gb=pd.DataFrame(cons_eng.groupby(cons_eng['Date'].dt.strftime('%Y-%m'))['Consommation'].mean()).sort_values(by='Date')  
cons_eng_gb
```

	Date	Consommation
0	2020-12	63700.449597
1	2020-11	55049.665972

# Représentation graphique de la consommation



- on remarque une forte saisonnalité
- mais qui semble liés au données de températures pics en période hivernale et d'été creux en périodes estivales
- on doit corriger de l'effet de température

# Données Dju

pour correction de l'effet de température

- DJU ( Degré Jour Unifié)
- valeur représentative de l'écart entre la **température** d'une journée et un **seuil** de température préétabli (18 °C ).
- Sommés sur une période,
- calcule les besoins de **chauffage et de climatisation d'un bâtiment.**

CALCUL DES DJU

Étape 1 ✓ Étape 2 Résultats

2. Sélectionnez la méthode de calcul (en savoir plus sur les méthodes de calculs)

☒ Météo ☐ Professionnels de l'énergie

3. Sélectionnez le type d'usage

☒ Chauffage ☒ Climatisation

4. Sélectionnez la température de référence

16° 17° 18° 19° 20° 21° 22° 23° 24° 25° 26° 27° 28° 29° 30°

5. Période de chauffage

Date de début 01/01/09 Date de fin 15/06/21

Les DJU sont proposés sur la période du 01/01/2009 au 15/06/2021.

☐ J'accepte les conditions d'utilisation de l'outil

Lancer la simulation >

Version 2.0.0

<https://cegibat.grdf.fr/simulateur/calcul-dju>

- chef lieu de région : **Paris Montessori**
- Période: **01/01/09 au 30/05/2021**

## Données Dju

- on retrouve des données mensuelles en colonnes
- et annuelles en lignes
- pour climatisation et chauffage
- de 2009-2020

## Données chauffage

chauf

	JAN	FÉV	MAR	AVR	MAI	JUN	JUI	AOÛ	SEP	OCT	NOV	DÉC
2020	339.0	249.6	268.6	81.4	65.7	20.6	0.9	4.5	34.3	157.5	227.2	336.8
2019	404.9	268.3	233.1	168.5	117.9	24.4	0.0	1.7	26.7	133.7	282.6	327.3

## Données sur la climatisation

clim

	JAN	FÉV	MAR	AVR	MAI	JUN	JUI	AOÛ	SEP	OCT	NOV	DÉC
2020	0	0	0	11.5	28.6	57.5	96.6	157.0	72.3	0.0	0.0	0
2019	0	0	0	5.7	3.2	100.1	152.3	115.6	22.6	4.0	0.0	0

df\_ch

	mois	chaufage
0	2020-01	339.0
1	2020-02	249.6

df\_clm

	mois	climatisation
0	2020-01	0.0
1	2020-02	0.0

## Jointure des données :

- on joints les dataframes de besoins en consommation , en chauffage, en climatisation
- un seul dataframe ayant la date en index
- on rajouter une colonne dju (clim et chauf)

```
elec=cons_eng_ren.merge(dju,how='inner').sort_  
elec
```

	mois	Consommation	chauffage	climatisation
107	2012-01	68261.475806	336.0	0.0
106	2012-02	77847.751437	435.9	0.0
105	2012-03	57722.497984	201.9	0.0

```
#mettre la colonne en dateTime  
elec['mois']=pd.to_datetime(elec['mois'])  
elec
```

	mois	Consommation	chauffage	climatisation
107	2012-01-01	68261.475806	336.0	0.0
106	2012-02-01	77847.751437	435.9	0.0

```
elec.isna().sum()
```

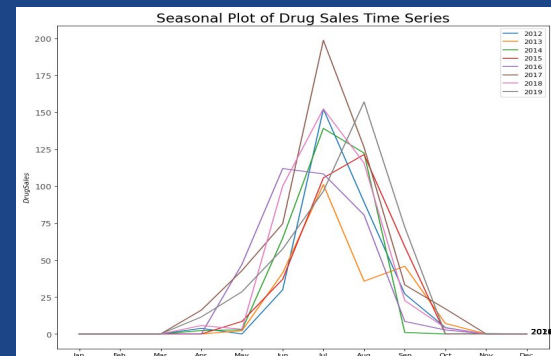
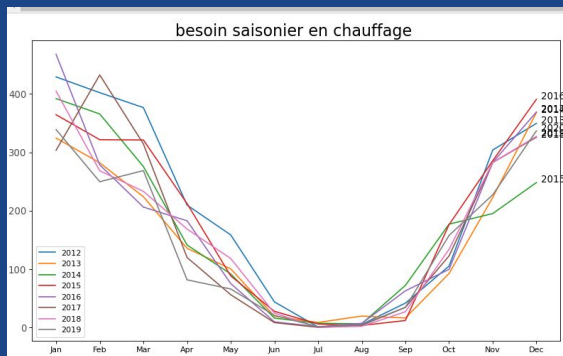
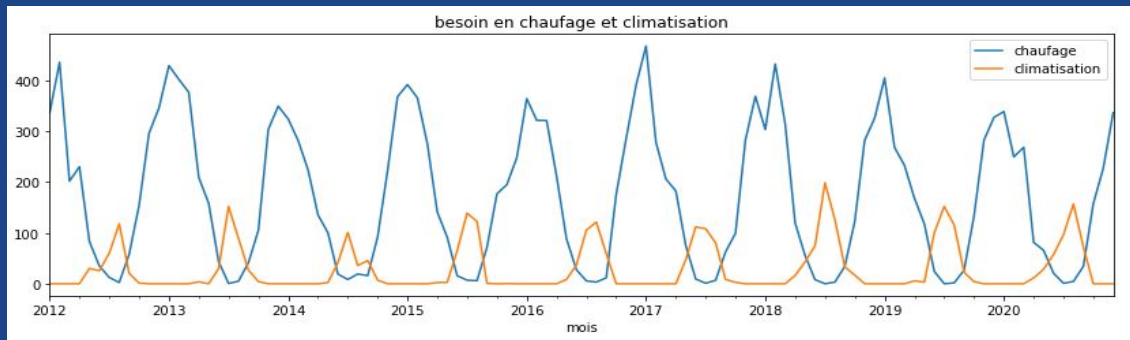
```
mois          0  
Consommation  0  
chauffage     0  
climatisation 0  
dtype: int64
```

```
elec.isnull().sum()
```

```
mois          0  
Consommation  0  
chauffage     0  
climatisation 0  
dtype: int64
```

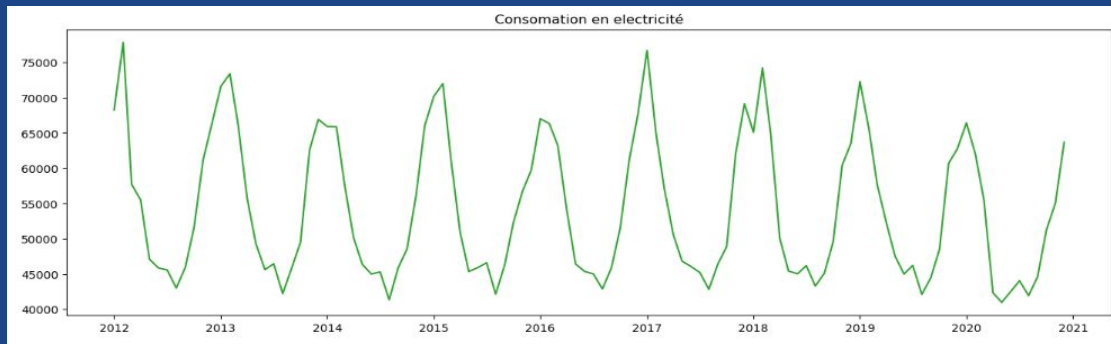
# Représentation graphique de la DJU

- les deux suivent deux saisonnalité différentes mais à dominance hivernale

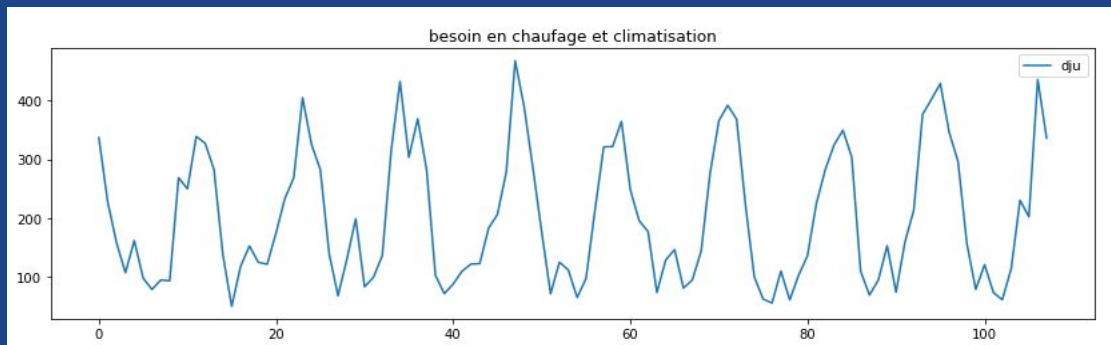




# Représentation graphique de la DJU et consommation d'énergie



- graphes de la consommation et du dju
- les deux suivent presque la même saisonnalité



la variation de la courbe **consommation** d'énergie sont effectivement liées aux **écarts de la température**

on suppose que les effets de température influe sur la consommation :

- ```
reg_e = smf.ols('Consommation~chaufage+climatisation', data=electr).fit()
print(reg_e.summary())
```
- OLS Regression Results
- |                   |                  |                     |          |
|-------------------|------------------|---------------------|----------|
| Dep. Variable:    | Consommation     | R-squared:          | 0.961    |
| Model:            | OLS              | Adj. R-squared:     | 0.961    |
| Method:           | Least Squares    | F-statistic:        | 1305.    |
| Date:             | Tue, 05 Jul 2022 | Prob (F-statistic): | 6.88e-75 |
| Time:             | 10:47:43         | Log-Likelihood:     | -971.47  |
| No. Observations: | 108              | AIC:                | 1949.    |
| Df Residuals:     | 105              | BIC:                | 1957.    |
| Df Model:         | 2                |                     |          |
| Covariance Type:  | nonrobust        |                     |          |
- |               | coef      | std err | t      | P> t  | [0.025  | 0.975   |
|---------------|-----------|---------|--------|-------|---------|---------|
| Intercept     | 4.001e+04 | 495.621 | 80.733 | 0.000 | 3.9e+04 | 4.1e+04 |
| chaufage      | 76.8834   | 1.901   | 40.437 | 0.000 | 73.113  | 80.653  |
| climatisation | 33.9385   | 5.731   | 5.922  | 0.000 | 22.575  | 45.301  |

# Tester la performance du modèle

## Verifier la colinearité des variables

```
variables = reg_e.model.exog  
[variance_inflation_factor(variables, i) for i in np.arange(1,variables.shape[1])]  
  
[1.95445692253576, 1.95445692253576]
```

- le test Pagan: **p-value < 5%**
- non rejet de **H0**
- les variances sont **constantes**

- les coefficients **VIF < 10**  
=> les variables ne **sont pas**  
**corrélées** entre elles

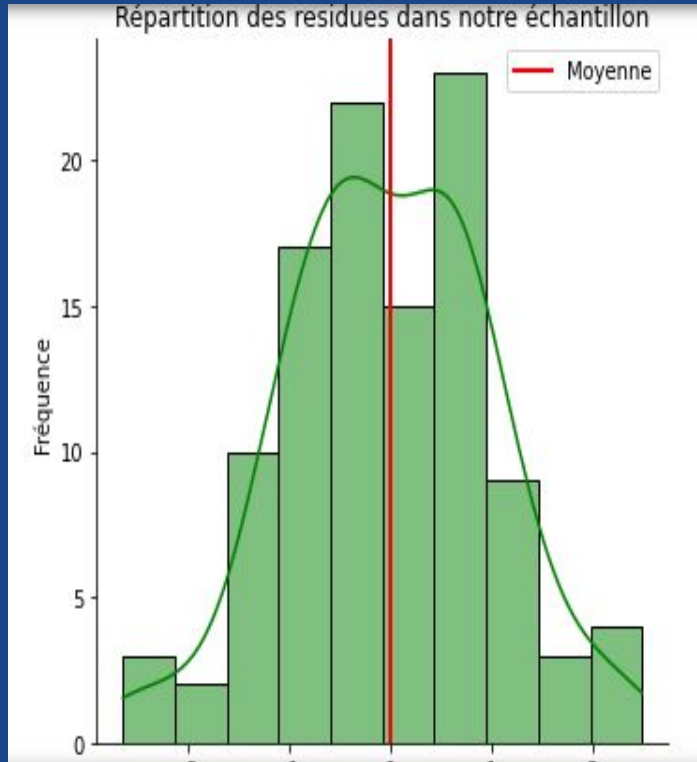
## tester l'homocidacité

```
#calcule des residues  
electr['rstudent'] = reg_e.get_influence().resid_
```

---

```
#tester l'homoscédasticité (c'est-à-dire la const  
_, pval, __, f_pval = statsmodels.stats.diagnosti  
print('p value test Breusch Pagan:', pval)  
  
p value test Breusch Pagan: 0.5772346011834666
```

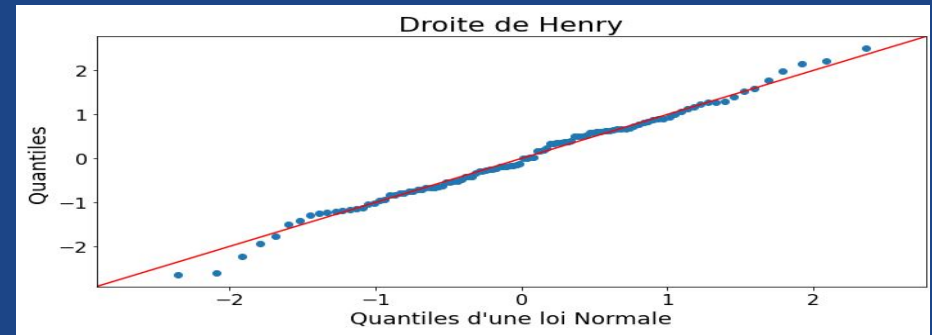
# Tester la performance du modèle: tester la normalité des résidus



- une distribution presque **symétrique** allure d'une **distribution normale**
- alignée avec la distribution théorique d'une loi normale.
- Un test de Shapiro-Wilk :  $p\text{-value} > 5\%$

les résidues suivent une loi normale

```
ShapiroResult(statistic=0.9925020337104797, pvalue=0.8224667310714722)
```



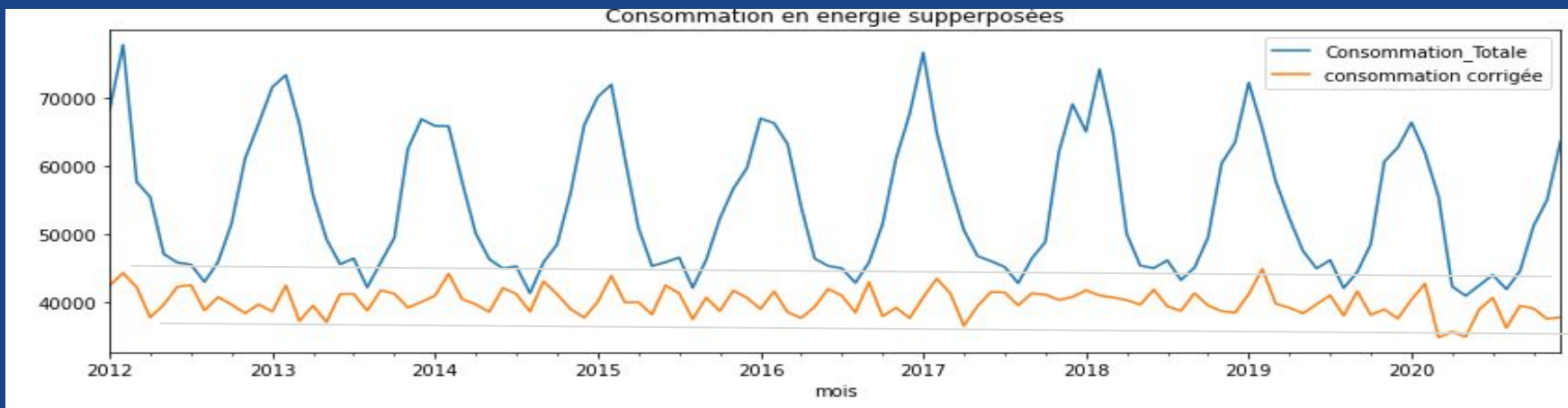
# Correction des données de consommation mensuelles de l'effet température:

- le modèle est validé
- correction avec les paramètres:
- $b1=76.88$ ,  $C1=33.94$

$$\text{conso}(\text{adj}) = \text{conso} - b1 * \text{chauf} - c1 * \text{clim}$$

```
electr['Consommation_corg1'] = electr['Consommation'] - (electr['chauffage']*b1)-(electr['climatisation']*c1)
```

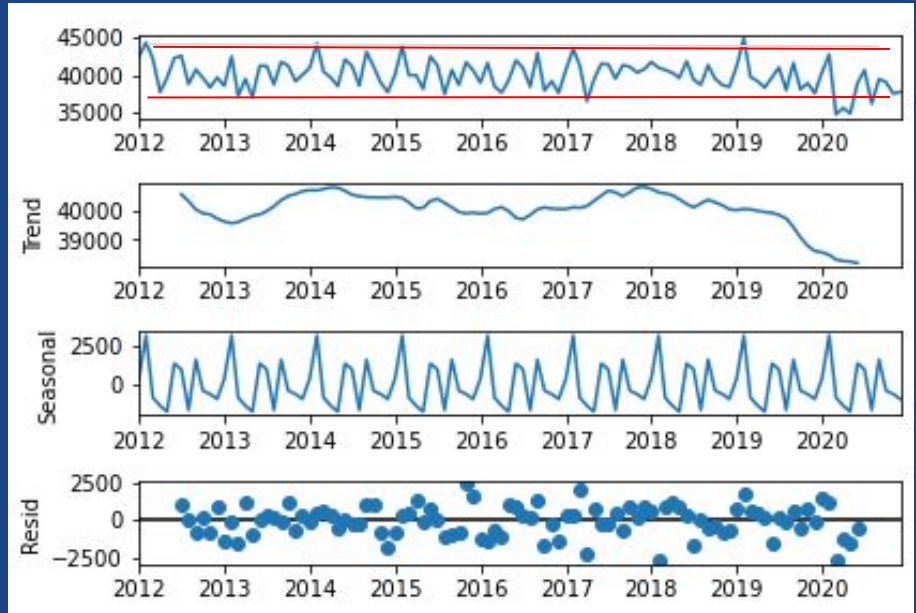
|            | Consommation | chauffage | climatisation | Consommation_corg1 |
|------------|--------------|-----------|---------------|--------------------|
| mois       |              |           |               |                    |
| 2012-01-01 | 68261.475806 | 336.0     | 0.0           | 42428.655018       |
| 2012-02-01 | 77847.751437 | 435.9     | 0.0           | 44334.279468       |



# décomposition de la série consommation corrigée

série est de type: **additive**

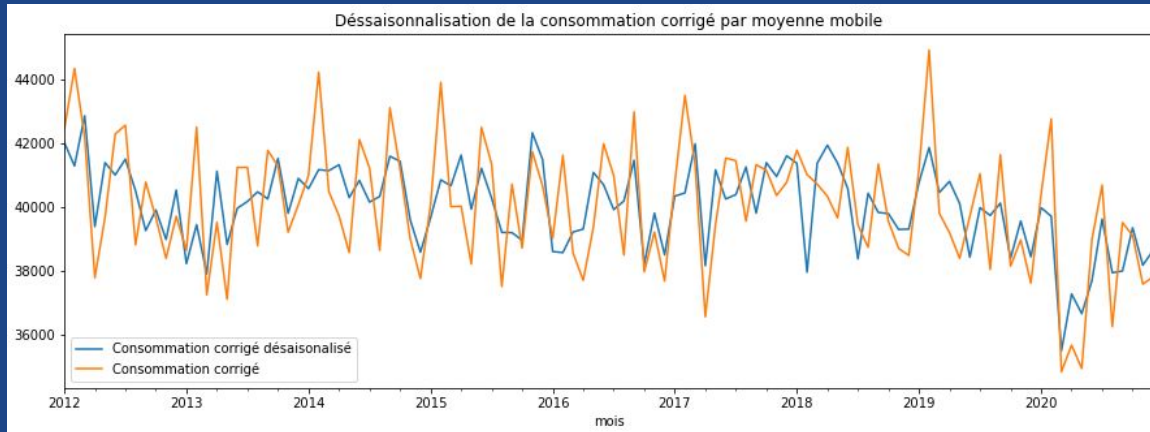
- la **tendance** est presque invariante
- saisonnalité de période de 12
- résidues faible
- on doit corriger la série des variations saisonnières



# correction des variation saisonnières de la consommation corrigée:

lissage triple de holt winters:il rajout la tendance et la saisonnalité

```
deseasonal = ser_corrig.values - result_mul.seasonal
```

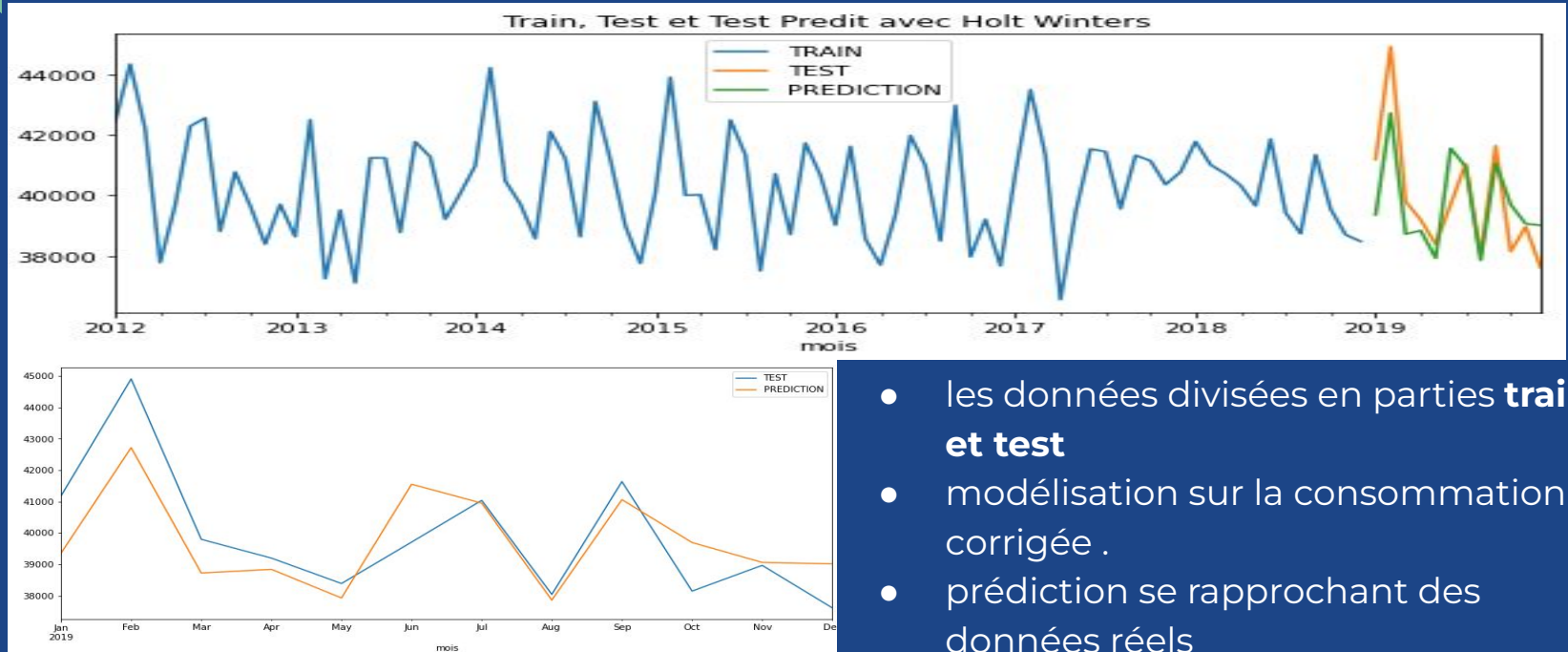


| mois       |              |
|------------|--------------|
| 2012-01-01 | 42015.795134 |
| 2012-02-01 | 41282.543698 |
| 2012-03-01 | 42854.827499 |
| 2012-04-01 | 39381.663164 |
| 2012-05-01 | 41386.567323 |
| ...        |              |
| 2020-08-01 | 37938.878863 |
| 2020-09-01 | 37990.910451 |
| 2020-10-01 | 39347.662840 |
| 2020-11-01 | 38173.633015 |
| 2020-12-01 | 38632.157253 |

- l'effet de la variation saisonnières est atténué pas de pics francs ni valeurs aberrantes
- pics positifs et négatifs périodique(12)
- on utilise ce modèle pour la **prédiction**



# Prévision sur la consommation méthode Holt Winters



Mean Absolute Error = 970.7395250858875  
Mean Squared Error = 1481834.3263622115

- les données divisées en parties **train** et **test**
- modélisation sur la consommation corrigée .
- prédiction se rapprochant des données réels
- evaluation métrique:faible



# Prévision sur la consommation méthode SARIMA

## données utilisées:

- pour avoir un modèle performant , méthode **SARIMA** plus puissant,
- pour la partie differentiation sur tout le data
- pour la partie training 12mois tronqués
- le test sur les 12 derniers mois

| Consommation_corg1 |              |
|--------------------|--------------|
| mois               |              |
| 2012-01-01         | NaN          |
| 2012-02-01         | 1905.624450  |
| 2012-03-01         | -2134.538975 |
| 2020-11-01         | -1522.600625 |
| 2020-12-01         | 224.363510   |

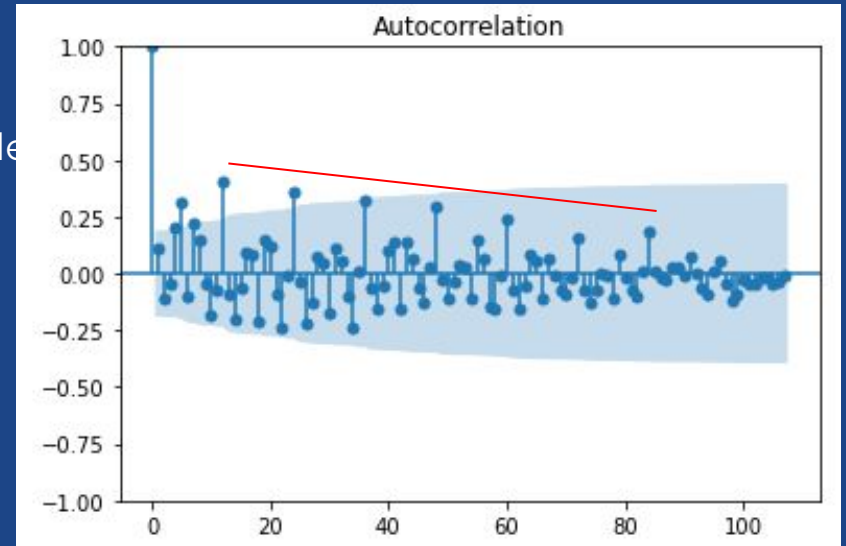
| Consommation_corg1 |              |
|--------------------|--------------|
| mois               |              |
| 2012-01-01         | 42428.655018 |
| 2012-02-01         | 44334.279468 |
| 2012-03-01         | 42199.740492 |
| 2018-11-01         | 38700.500577 |
| 2018-12-01         | 38478.891691 |

| Consommation_corg1 |              |
|--------------------|--------------|
| mois               |              |
| 2019-01-01         | 41178.286266 |
| 2019-02-01         | 44910.825692 |
| 2019-03-01         | 39796.498723 |
| 2019-11-01         | 38965.904599 |
| 2019-12-01         | 37610.153460 |

# Prévision sur la consommation méthode SARIMA

- pour avoir un modèle performant , méthode **SARIMA** plus puissant,
- La sortie ACF présente une décroissance lente vers 0, ce qui traduit un problème de non-stationnarité
- tester la **stationnarité** de notre série avec le **test ADF** (Augmented Dickey-Fuller).
- **non rejet de** l'hypothèse de non-stationnarité de la série,
- **contrairement** à la série **différenciée**.

donc **On effectue donc une différenciation ( $I-B$ )**



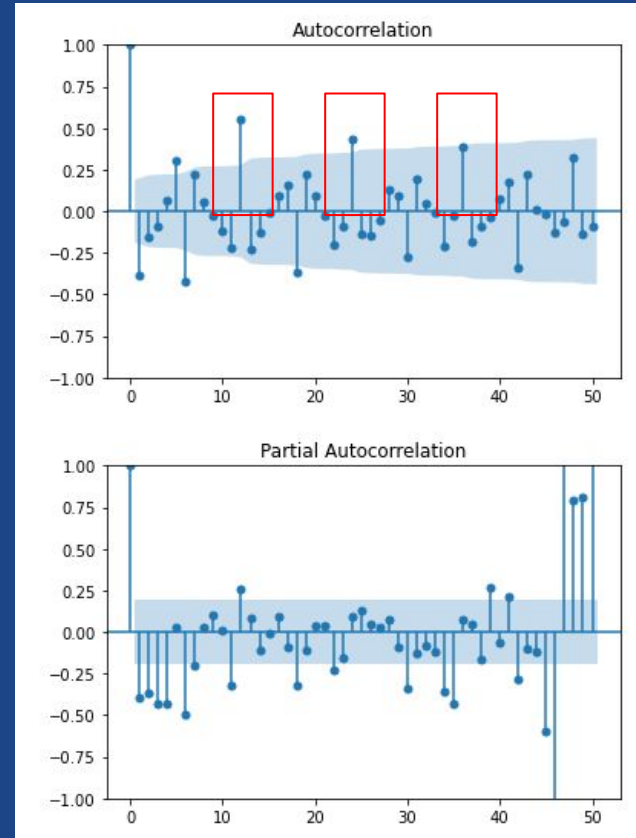
p-value ADF série originelle : 0.27031869816536236  
p-value ADF série différenciée : 0.005027668770853414

# Prévision sur la consommation méthode SARIMA

## différenciation à l'ordre (I-B)

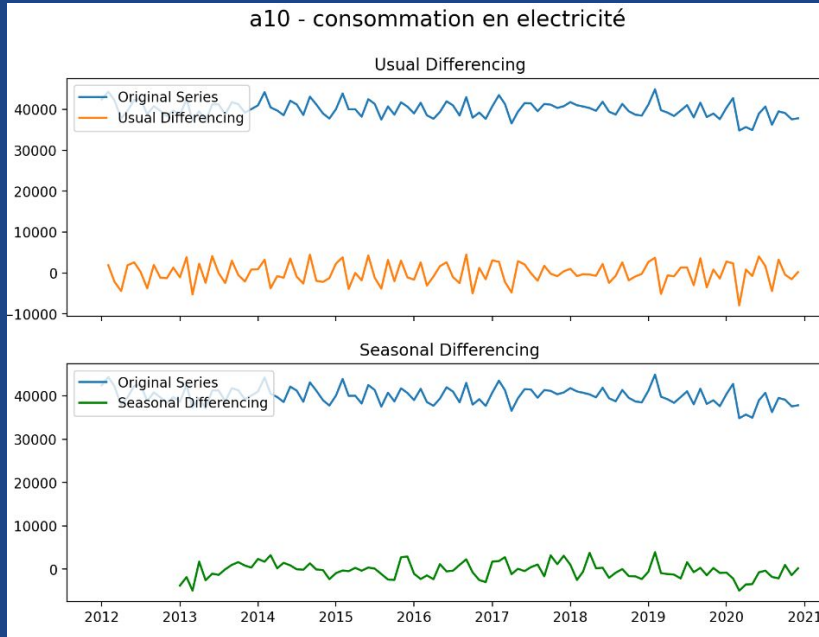
- Des pics significatifs dans le graphique ACF au décalage 6, 12, 18... (composante saisonnière semestrielle).
- Dans le graphique PACF, des pics positifs et négatifs significatifs.
- série non stationnaire

donc **On effectue une deuxième différenciation d'ordre** donc une **(I-B12)** pour éliminer la saisonnalité

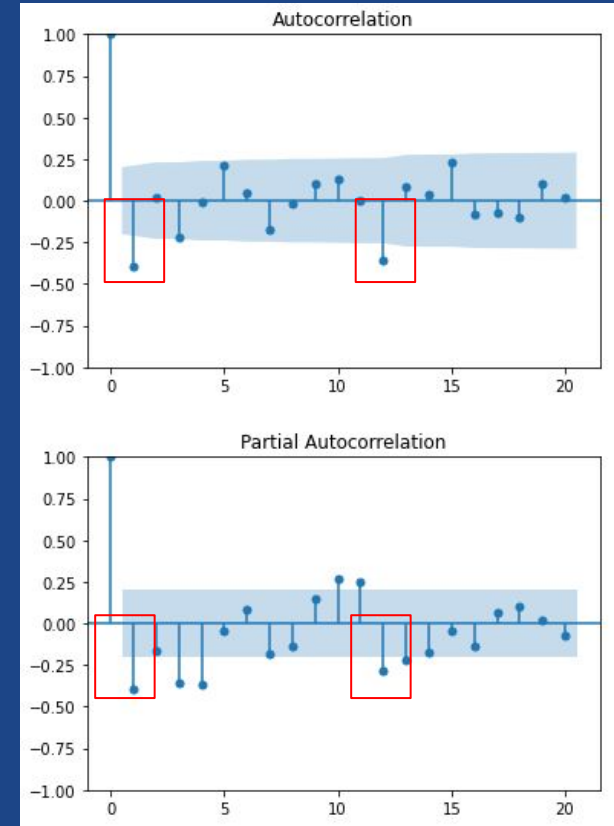


# Prévision sur la consommation méthode SARIMA

## différenciation à l'ordre (I-B12)



- la stationnarité semble être établie
- on passe à Déterminer des ordres optimaux



# Prévision sur la consommation méthode SARIMA

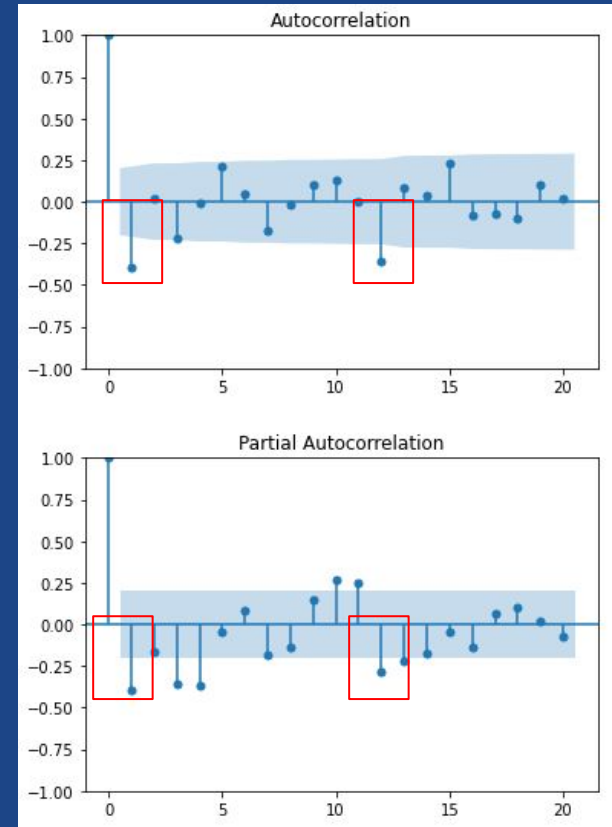
## différenciation à l'ordre (I-B12)

- autocorrelation simples et partielle s'annulent a partir du rang  $p+1$
- on suppose que  $c'$  est un model

$(p,d,q),(P,D,Q)$

ARIMA(1,1,1),(1,1,1)12

Estimer les paramètres et les départager par l'AIC (ou le BIC)



# Prévision sur la consommation méthode SARIMA

Estimer les paramètres du model 1  $\text{ARIMA}(1, 1, 1), (1, 1, 1)_{12}$

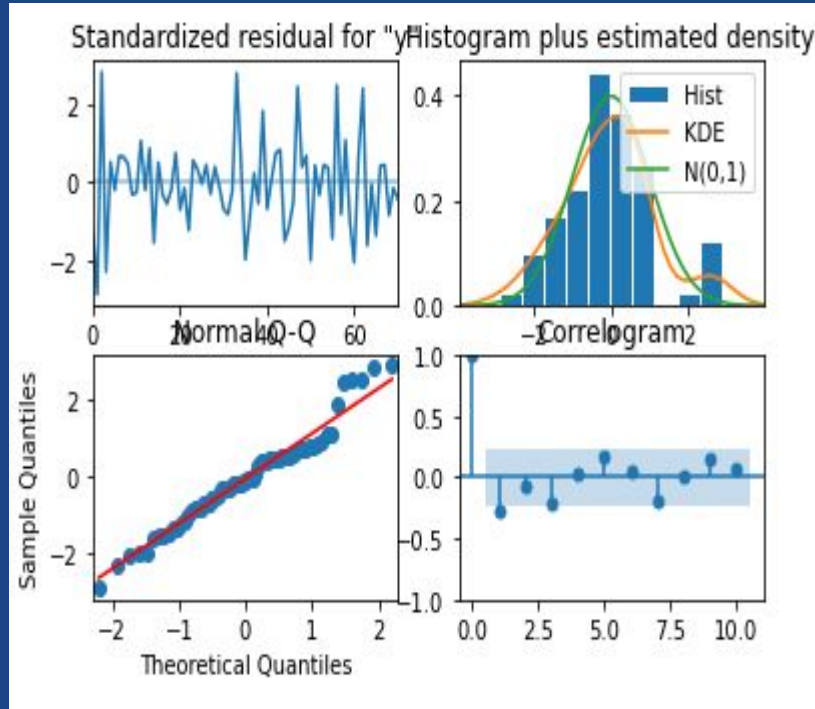
| SARIMAX Results         |                                |                   |         |       |           |          |
|-------------------------|--------------------------------|-------------------|---------|-------|-----------|----------|
| Dep. Variable:          | y                              | No. Observations: | 8       |       |           |          |
| Model:                  | SARIMAX(1, 1, 1)x(1, 1, 1, 12) | Log Likelihood    | -644.83 |       |           |          |
| Date:                   | Thu, 07 Jul 2022               | AIC               | 1299.67 |       |           |          |
| Time:                   | 19:05:37                       | BIC               | 1310.99 |       |           |          |
| Sample:                 | 0                              | HQIC              | 1304.17 |       |           |          |
|                         | - 84                           |                   |         |       |           |          |
| Covariance Type:        | opg                            |                   |         |       |           |          |
|                         | coef                           | std err           | z       | P> z  | [0.025    | 0.975]   |
| ar.L1                   | 0.3851                         | 0.146             | 2.637   | 0.008 | 0.099     | 0.671    |
| ma.L1                   | -0.5442                        | 0.118             | -4.601  | 0.000 | -0.776    | -0.312   |
| ar.S.L12                | -0.7080                        | 558.606           | -0.001  | 0.999 | -1095.555 | 1094.139 |
| ma.S.L12                | 0.7080                         | 558.633           | 0.001   | 0.999 | -1094.193 | 1095.609 |
| sigma2                  | 3.203e+06                      | 4.19e+05          | 7.647   | 0.000 | 2.38e+06  | 4.02e+06 |
| Ljung-Box (L1) (Q):     | 5.77                           | Jarque-Bera (JB): | 1.52    |       |           |          |
| Prob(Q):                | 0.02                           | Prob(JB):         | 0.47    |       |           |          |
| Heteroskedasticity (H): | 1.18                           | Skew:             | 0.28    |       |           |          |
| Prob(H) (two-sided):    | 0.70                           | Kurtosis:         | 3.44    |       |           |          |

```
6 : lb_stat      1.415924
lb_pvalue      0.759581
dtype: float64
12 : lb_stat      4.360298
lb_pvalue      0.775777
dtype: float64
18 : lb_stat     12.844748
lb_pvalue      0.522819
dtype: float64
24 : lb_stat     17.308334
lb_pvalue      0.416067
dtype: float64
30 : lb_stat     20.072889
lb_pvalue      0.391080
dtype: float64
36 : lb_stat     22.027058
lb_pvalue      0.417585
```

- la p-value RA.12;MA.S.L12 (partie saisonnière) Q n'est pas significative
- les indicateurs de performance AIC et BIC semblent faible
- le **test de blancheur** de jung Box :les p values non significatives => indiquent un bruit blanc

# Prévision sur la consommation méthode SARIMA

## diagnostic des résidus



```
#Test de Jarque-Bera  
results0.test_normality("jarquebera")[0][1]
```

0.46795658962724085

- Test utilisé: **Jarque-Bera** ne prend pas en compte les premiers résidus qui **divergent** légèrement d'un schéma gaussien
- le test non significatif donc
- non rejet de l'hypothèse de la **normalité des résidues**

on enlève la partie RA.12 MA(12) au model et on test a nouveau



# Prévision sur la consommation méthode SARIMA

Estimer les paramètres sur le modèle amélioré **(1,1,1),(0,1,0,12)**

```
SARIMAX Results
=====
Dep. Variable:          y      No. Observations:      84
Model:                SARIMAX(1, 1, 1)x(0, 1, [], 12)  Log Likelihood: -643.455
Date:                 Thu, 07 Jul 2022                AIC: 1292.910
Time:                 19:16:16                        BIC: 1299.698
Sample:               0                               HQIC: 1295.610
Covariance Type:      opg
=====
              coef    std err          z      P>|z|      [0.025    0.975]
-----
ar.L1         0.3721     0.202       1.841     0.066     -0.024     0.768
ma.L1        -0.5135     0.166      -3.089     0.002     -0.839    -0.188
sigma2        4.224e+06   6.52e+05   6.483     0.000   2.95e+06   5.5e+06
=====
Ljung-Box (L1) (Q):                6.19   Jarque-Bera (JB):                1.81
Prob(Q):                          0.01   Prob(JB):                      0.40
Heteroskedasticity (H):            1.19   Skew:                          0.32
Prob(H) (two-sided):              0.67   Kurtosis:                     3.46
=====
```

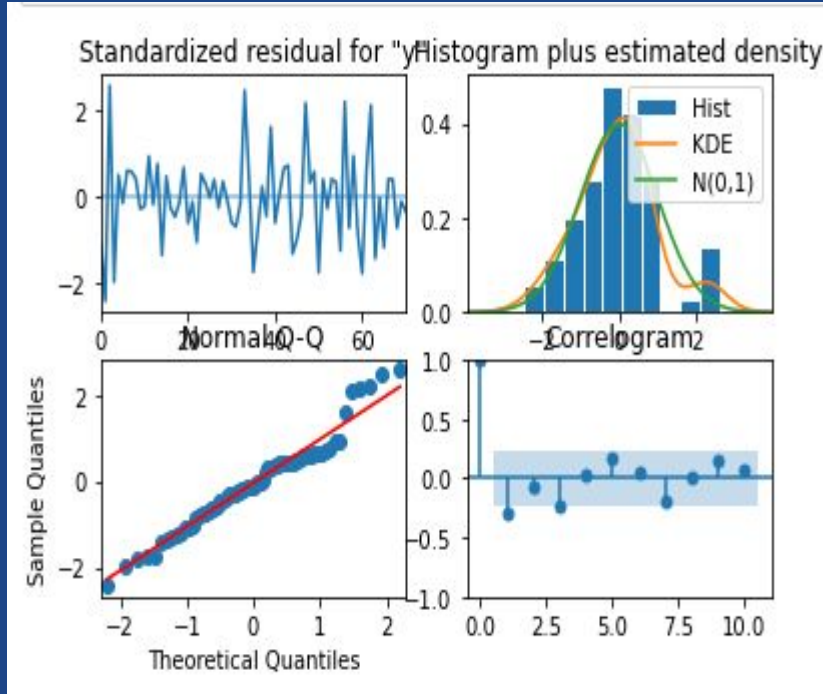
```
[1] Covariance matrix calc
6 : lb_stat      1.371286
lb_pvalue      0.772174
dtype: float64
12 : lb_stat      4.334899
lb_pvalue      0.783224
dtype: float64
18 : lb_stat     12.89830
lb_pvalue      0.527459
dtype: float64
24 : lb_stat     17.39745
lb_pvalue      0.418589
dtype: float64
30 : lb_stat     20.18118
lb_pvalue      0.391573
dtype: float64
36 : lb_stat     22.14826
lb_pvalue      0.416470
dtype: float64
```

- les paramètres sont non significatifs AR.1
- les indicateurs de performance **AIC et BIC** semblent plus **faibles**
- le **test de blancheur** de jung Box :les p values non significatives => indiquent un bruit blanc



# Prévision sur la consommation méthode SARIMA

diagnostic des résidus sur le model 2:  $(1,1,1),(0,1,0,12)$



```
#Test de Jarque-Bera  
results1.test_normality("jarquebera")[0][1]
```

0.40462704569811303

- Test **Jarque-Bera** conclu au non rejet de l'hypothèse de la **normalité des résidues**
- **diagrammes** indiquent une **disposition semblable à celle normale**

on retire la partie MA(12) au model et on test a nouveau

# Prévision sur la consommation méthode SARIMA

Estimer les paramètres sur le modèle amélioré **(0,1,1),(0,1,0.12)**

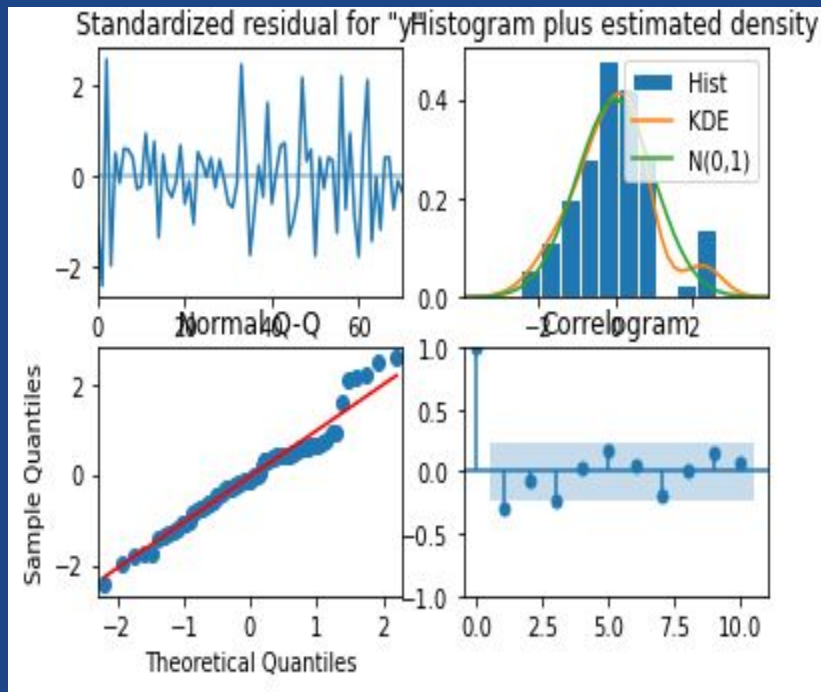
| SARIMAX Results         |                                 |                   |                   |       |          |          |
|-------------------------|---------------------------------|-------------------|-------------------|-------|----------|----------|
| =====                   |                                 |                   |                   |       |          |          |
| Dep. Variable:          | y                               | No. Observations: | 84                |       |          |          |
| Model:                  | SARIMAX(0, 1, 1)x(0, 1, [], 12) | Log Likelihood    | 643.499           |       |          |          |
| Date:                   | Thu, 07 Jul 2022                | AIC               | 1290.998          |       |          |          |
| Time:                   | 19:32:04                        | BIC               | 1295.524          |       |          |          |
| Sample:                 | 0                               | HQIC              | 1292.798          |       |          |          |
|                         | - 84                            |                   |                   |       |          |          |
| Covariance Type:        | opg                             |                   |                   |       |          |          |
| =====                   |                                 |                   |                   |       |          |          |
|                         | coef                            | std err           | z                 | P> z  | [0.025   | 0.975]   |
| -----                   |                                 |                   |                   |       |          |          |
| ma.L1                   | -0.1449                         | 0.050             | -2.870            | 0.004 | -0.244   | -0.046   |
| sigma2                  | 4.267e+06                       | 6.38e+05          | 6.691             | 0.000 | 3.02e+06 | 5.52e+06 |
| -----                   |                                 |                   |                   |       |          |          |
| Ljung-Box (L1) (Q):     |                                 | 4.64              | Jarque-Bera (JB): |       | 5.29     |          |
| Prob(Q):                |                                 | 0.03              | Prob(JB):         |       | 0.07     |          |
| Heteroskedasticity (H): |                                 | 1.31              | Skew:             |       | 0.57     |          |
| Prob(H) (two-sided):    |                                 | 0.51              | Kurtosis:         |       | 3.70     |          |
| =====                   |                                 |                   |                   |       |          |          |

```
6 : lb_stat      1.576410
lb_pvalue      0.735775
dtype: float64
12 : lb_stat      4.675704
lb_pvalue      0.755824
dtype: float64
18 : lb_stat     13.302670
lb_pvalue      0.508447
dtype: float64
24 : lb_stat     17.829830
lb_pvalue      0.401953
dtype: float64
30 : lb_stat     20.626130
lb_pvalue      0.374307
dtype: float64
36 : lb_stat     22.612870
lb_pvalue      0.397481
dtype: float64
```

- les paramètres sont significatifs
- les indicateurs de performance **AIC et BIC** semblent plus **faibles**
- le **test de blancheur** de jung Box : les p values non significatives => indiquent un bruit blanc

# Prévision sur la consommation méthode SARIMA

diagnostic des résidus sur le model 2: **(0,1,1),(0,1,0,12)**



```
#Test de Jarque-Bera
results2.test_normality("jarquebera")[0][1]
0.071153098748574
```

- Test **Jarque-Bera** conclu au non rejet de l'hypothèse de la **normalité des résidues**
- **diagrammes** indiquent une **disposition semblable à celle normale**

on peut retenir ce model

# Prévision sur la consommation méthode SARIMA

## SARIMA optimal

- Construction d'un modèle SARIMA en utilisant la fonction **auto arima()** de **pmdarima**,
- pour la fréquence  $m=12$ ,  $D=1$ .

|                |                                |                   |          |
|----------------|--------------------------------|-------------------|----------|
| Dep. Variable: | y                              | No. Observations: | 84       |
| Model:         | SARIMAX(1, 1, 1)x(0, 1, 1, 12) | Log Likelihood    | -634.241 |
| Date:          | Thu, 07 Jul 2022               | AIC               | 1276.482 |
| Time:          | 09:28:25                       | BIC               | 1285.532 |

le model choisi est le model **(1,1,1)(0,1,1)[12]**

```
Performing stepwise search to minimize aic
ARIMA(1,1,1)(0,1,1)[12] : AIC=1276.482, Time=0.32 sec
ARIMA(0,1,0)(0,1,0)[12] : AIC=1294.855, Time=0.04 sec
ARIMA(1,1,0)(1,1,0)[12] : AIC=1283.706, Time=0.10 sec
ARIMA(0,1,1)(0,1,1)[12] : AIC=1282.488, Time=0.15 sec
ARIMA(1,1,1)(0,1,0)[12] : AIC=1292.910, Time=0.26 sec
ARIMA(1,1,1)(1,1,1)[12] : AIC=1299.678, Time=0.25 sec
ARIMA(1,1,1)(0,1,2)[12] : AIC=1277.349, Time=0.82 sec
ARIMA(1,1,1)(1,1,0)[12] : AIC=1278.325, Time=0.32 sec
ARIMA(1,1,1)(1,1,2)[12] : AIC=inf, Time=1.53 sec
ARIMA(1,1,0)(0,1,1)[12] : AIC=1282.043, Time=0.13 sec
ARIMA(2,1,1)(0,1,1)[12] : AIC=1278.969, Time=0.38 sec
ARIMA(1,1,2)(0,1,1)[12] : AIC=1286.864, Time=0.41 sec
ARIMA(0,1,0)(0,1,1)[12] : AIC=1289.557, Time=0.10 sec
ARIMA(0,1,2)(0,1,1)[12] : AIC=1284.624, Time=0.19 sec
ARIMA(2,1,0)(0,1,1)[12] : AIC=1282.435, Time=0.16 sec
ARIMA(2,1,2)(0,1,1)[12] : AIC=1279.115, Time=0.56 sec
ARIMA(1,1,1)(0,1,1)[12] intercept : AIC=1278.386, Time=0.51 sec

Best model: ARIMA(1,1,1)(0,1,1)[12]
Total fit time: 6.266 seconds
```

# Prévision sur la consommation méthode SARIMA

optimal SARIMA **(1,1,1)(0,1,1)[12]**

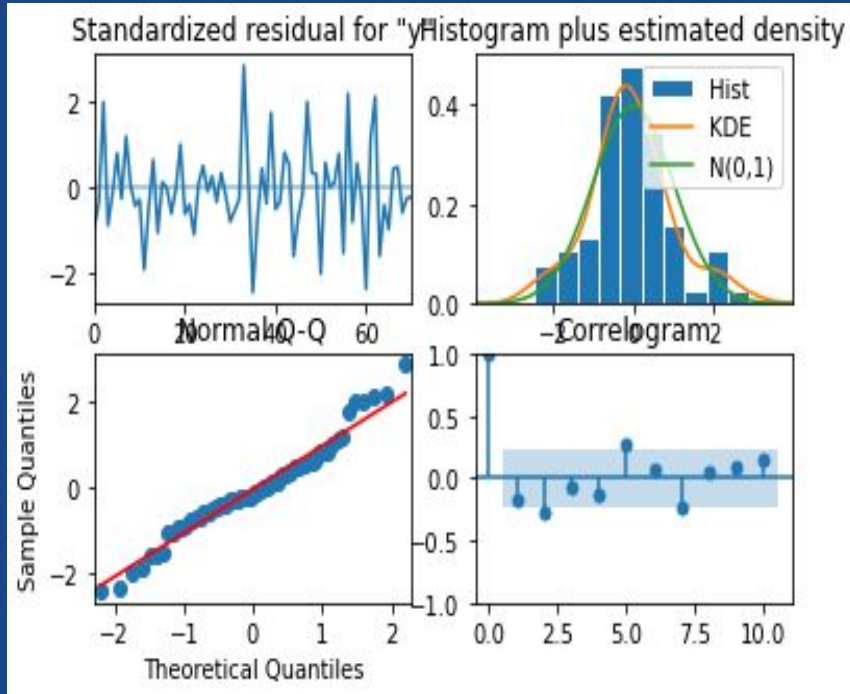
| SARIMAX Results         |                                |                   |          |       |         |          |
|-------------------------|--------------------------------|-------------------|----------|-------|---------|----------|
| Dep. Variable:          | y                              | No. Observations: | 84       |       |         |          |
| Model:                  | SARIMAX(1, 1, 1)x(0, 1, 1, 12) | Log Likelihood    | -634.241 |       |         |          |
| Date:                   | Thu, 07 Jul 2022               | AIC               | 1276.482 |       |         |          |
| Time:                   | 11:43:20                       | BIC               | 1285.532 |       |         |          |
| Sample:                 | 0                              | HQIC              | 1280.081 |       |         |          |
|                         | - 84                           |                   |          |       |         |          |
| Covariance Type:        | opg                            |                   |          |       |         |          |
|                         | coef                           | std err           | z        | P> z  | [0.025  | 0.975]   |
| ar.L1                   | -0.7977                        | 0.126             | -6.351   | 0.000 | -1.044  | -0.552   |
| ma.L1                   | 0.6500                         | 0.156             | 4.178    | 0.000 | 0.345   | 0.955    |
| ma.S.L12                | -0.2233                        | 0.041             | -5.445   | 0.000 | -0.304  | -0.143   |
| sigma2                  | 3.203e+06                      | 4.62e+05          | 6.939    | 0.000 | 2.3e+06 | 4.11e+06 |
| Ljung-Box (L1) (Q):     | 2.15                           | Jarque-Bera (JB): | 2.57     |       |         |          |
| Prob(Q):                | 0.14                           | Prob(JB):         | 0.28     |       |         |          |
| Heteroskedasticity (H): | 2.00                           | Skew:             | 0.31     |       |         |          |
| Prob(H) (two-sided):    | 0.10                           | Kurtosis:         | 3.70     |       |         |          |

```
Retard : p-value
6 : lb_stat      1.801166
lb_pvalue      0.665040
dtype: float64
12 : lb_stat     4.270485
lb_pvalue      0.715927
dtype: float64
18 : lb_stat     10.151397
lb_pvalue      0.521580
dtype: float64
24 : lb_stat     13.215083
lb_pvalue      0.493048
dtype: float64
30 : lb_stat     15.141654
lb_pvalue      0.536798
dtype: float64
36 : lb_stat     16.537049
lb_pvalue      0.596844
```

- les paramètres sont tous significatifs
- les indicateurs de performance **AIC et BIC** semblent plus **faible**
- le **test de blancheur** de jung Box :les p values non significatives sauf pour le retard 24 à la limite 0.49%. => indiquent un bruit blanc

# Prévision sur la consommation méthode SARIMA

tester la normalité des résidues du modèle optimal  $(1,1,1)(0,1,1)[12]$



```
#Test de Jarque-Bera
results3.test_normality("jarquebera")[0][1]
0.2771802880260108
```

- Test **Jarque-Bera** conclu au non rejet de l'hypothèse de la **normalité des résidues**
- **diagrammes** indiquent une **disposition semblable à celle normale**

on retire la partie MA(12) au model et on test a nouveau

# Prévision sur la consommation méthode SARIMA

## le choix du modèle OPTIMAL

**:(0,1,1),(0,1,0,12)**

|                |           |
|----------------|-----------|
| Log Likelihood | -8151.155 |
| AIC            | 1290.998  |
| BIC            | 1295.524  |

- paramètre non significatif
- identification du bruit blanc
- résidues suivent loi normale

**(1,1,1)(0,1,1)[12]**

**AIC 1276.482**

**BIC 1285.532**

- paramètre non significatif
- identification du bruit blanc
- résidues suivent loi normale



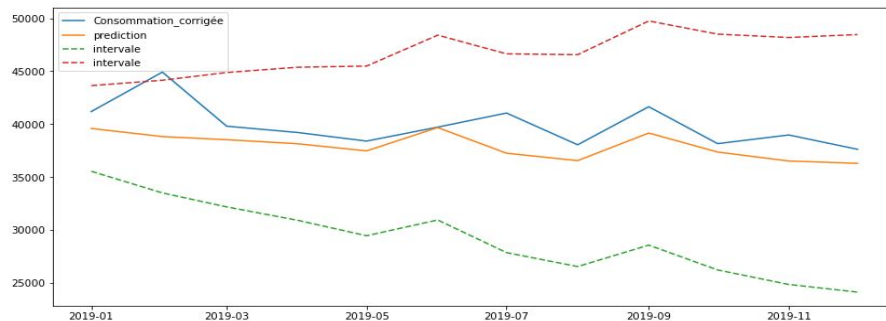
# Analyse a posteriori

## Prévision à partir des deux modèles

- prévision sur les deux modèles
- les deux modèles ont sortis des prédictions comprises dans l'intervalle de **prévision à 95%**
- **se rapprochent** des données du test

le choix du model 3 :

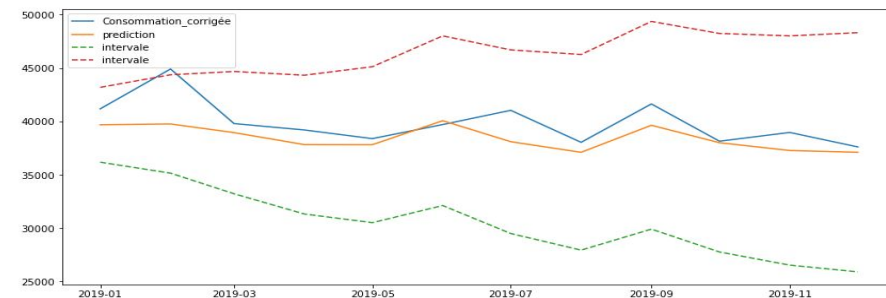
- résultats qui se rapprochant des données test
- MAE et MSE plus faible



évaluation métrique du model Sarima model2:  $(1,1,1),(1,1,0,12)$

```
from sklearn.metrics import mean_absolute_error, mean_squared_error
print(f'Mean Absolute Error = {mean_absolute_error(t_s, p_s)}')
print(f'Mean Squared Error = {mean_squared_error(t_s, p_s)}')
```

Mean Absolute Error = 1947.4581670816303  
Mean Squared Error = 6222198.237265994



évaluation métrique du model Sarima model 3:  $(1,1,1)(0,1,1)[12]$

```
from sklearn.metrics import mean_absolute_error, mean_squared_error
print(f'Mean Absolute Error = {mean_absolute_error(t_s, p_s)}')
print(f'Mean Squared Error = {mean_squared_error(t_s, p_s)}')
```

Mean Absolute Error = 1499.717676893689  
Mean Squared Error = 4037980.9137362097

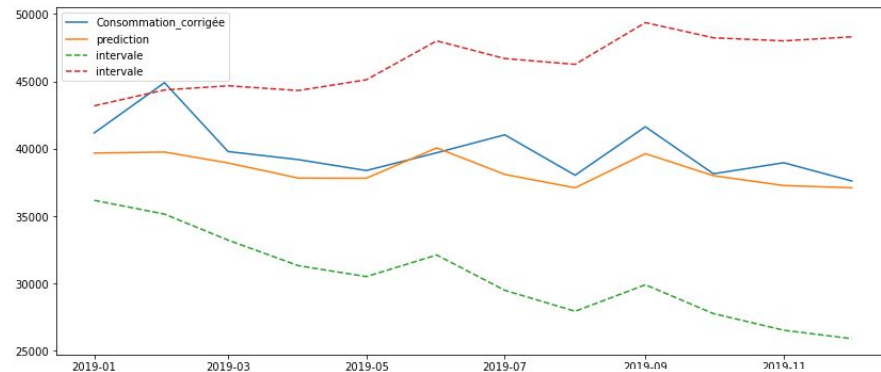


# Analyse a posteriori

## comparaison méthode holt-winters et SARIMA

- la méthode de holt winter semble être plus proche des données
- avec une métrique plus faible que celle de SARIMA

il y existe un modèle plus performant que le modèle  $(1,1,1),(0,1,1,12)$   
**comparer avec map**

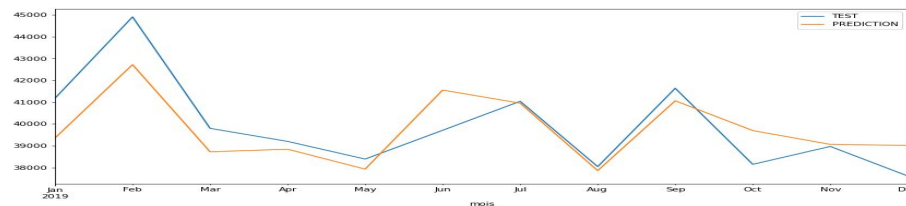


model 3:  $(1,1,1)(0,1,1)[12]$

### évaluation métrique du model Sarima

```
from sklearn.metrics import mean_absolute_error, mean_squared_error
print(f'Mean Absolute Error = {mean_absolute_error(t_s, p_s)}')
print(f'Mean Squared Error = {mean_squared_error(t_s, p_s)}')
```

Mean Absolute Error = 1499.717676893689  
Mean Squared Error = 4037980.9137362097



model: **holt\_winters**

Mean Absolute Error = 970.7395250858875  
Mean Squared Error = 1481834.3263622115



## Conclusion:

- 01 les deux methodes de prevision sont satisfaisantes se rapprochent des données du test
- 02 un meilleur modèle peut être obtenue avec SARIMA