Projet 9

Prédisez la demande en électricité

Predict electricity demand

Notre objectif

En tant qu'employé d'Enercoop, une entreprise spécialisée dans les énergies renouvelables, nous devons trouver un moyen de prévoir la demande afin de pouvoir fournir une offre adéquate.

Notre jeu de données

Nos données :

- La consommation mensuelle d'électricité de rte-france.com

- Données de température de GRDF

Mois	Qualité	Territoire	Production totale	Production nucléaire	Production thermique totale	Production thermique charbon	Production thermique fioul	Production thermique gaz
2012-01-01	Données définitives	France	57177	42811.0	5399	741.0	691	3967
2012-02-01	Données définitives	France	54419	38661.0	8721	2511.0	1309	4901
2012-03-01	Données définitives	France	48583	37549.0	5276	1435.0	666	3175
2012-04-01	Données définitives	France	44192	33100.0	3484	1655.0	486	1343
2012-05-01	Données définitives	France	40433	29058.0	1772	854.0	368	549

	JAN	FÉV	MAR	AVR	MAI	JUN	JUI	AOÛ	SEP	OCT	NOV	DÉC	Total
2021	396.7	302.8	271.0	228.3	138.3	1.4	0.0	0.0	0.0	0.0	0.0	0.0	1338.2
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	1785.9
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	1989.0
2018	303.4	432.6	314.3	119.7	55.9	8.1	0.0	3.3	34.3	122.4	282.5	325.9	2002.2
2017	467.9	278.4	206.1	182.6	75.0	9.4	1.0	6.8	62.6	99.4	282.6	369.0	2040.6
2016	364.4	321.6	321.1	212.1	88.1	27.5	5.7	3.2	11.7	176.0	285.6	390.8	2207.3
2015	392.0	365.7	275.5	141.1	91.5	15.8	6.9	6.1	71.9	176.9	195.0	248.1	1986.2
2014	324.4	281.9	223.9	135.5	100.2	19.1	8.3	19.3	16.0	92.3	222.6	368.2	1811.5
2013	429.2	402.2	376.6	209.5	158.4	43.6	0.6	5.0	41.5	105.0	303.9	349.5	2424.8
2012	336.0	435.9	201.9	230.3	83.3	35.0	12.4	2.4	58.0	154.6	296.2	345.9	2191.5
2011	392.0	304.8	243.1	77.6	43.4	31.4	15.0	11.9	23.2	127.6	226.6	312.7	1809.0
2010	499.2	371.4	294.5	165.3	140.9	22.6	0.0	11.1	52.3	172.2	310.0	512.0	2551.1
2009	486.8	365.7	293.2	135.1	82.2	39.8	3.1	0.9	26.9	149.6	224.7	411.8	2219.7

Notre jeu de données

```
1 print((df.isna().sum()/df.shape[0]*100).round(2))
2 print('')
3 print(df.duplicated().sum())
```

```
Qualité                                    0.00
Territoire                                 0.00
Production totale                          0.00
Production nucléaire                      34.93
Production thermique totale                0.00
Production thermique charbon              30.58
Production thermique fioul                 0.00
Production thermique gaz                   0.00
Production hydraulique                     0.00
Production éolien                          0.00
Production solaire                         0.00
Production bioénergies                     0.00
Consommation totale                        0.00
Solde exportateur                          5.99
Echanges export                           91.30
Echanges import                           91.30
Echanges avec le Royaume-Uni              91.30
Echanges avec l'Espagne                   91.30
Echanges avec l'Italie                    91.30
Echanges avec la Suisse                   91.30
Echanges avec l'Allemagne et la Belgique  92.44
dtype: float64

0
```

```
1 df.Territoire.value_counts()
```

```
France                      119
Grand-Est                   107
Hauts-de-France             107
Normandie                   107
Occitanie                   107
Bretagne                    107
Bourgogne-Franche-Comté     107
Auvergne-Rhône-Alpes        107
Nouvelle-Aquitaine          107
Ile-de-France               107
Pays-de-la-Loire            107
Centre-Val de Loire         107
PACA                        107
Name: Territoire, dtype: int64
```

Grand pourcentage de valeurs manquantes dans certaines colonnes, mais elles ne semblent pas importantes. Le cadre de données n'a pas de valeurs dupliquées.

La dataframe est divisée en régions, mais contient également des données globales pour l'ensemble du pays (France).

Notre jeu de données

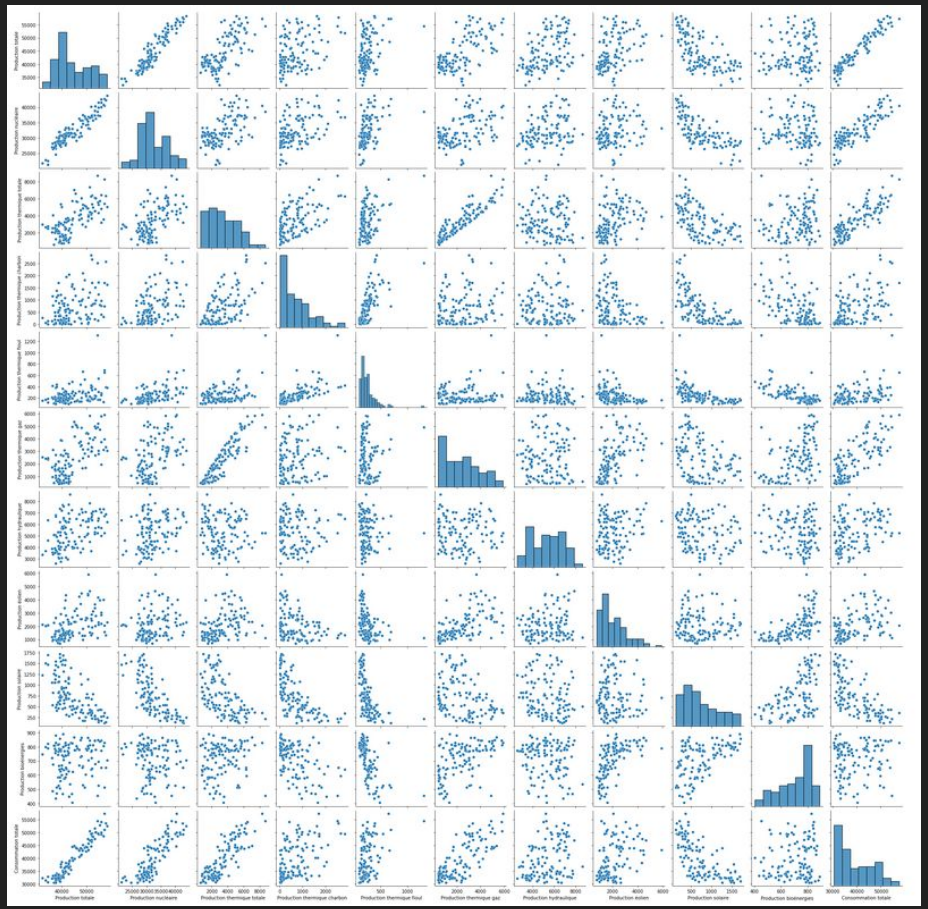
```
# supprimer les régions, consolider les données en tant que France seulement
# remove regions, consolidate data as France only

df = df.loc[df['Territoire'] == 'France']
```

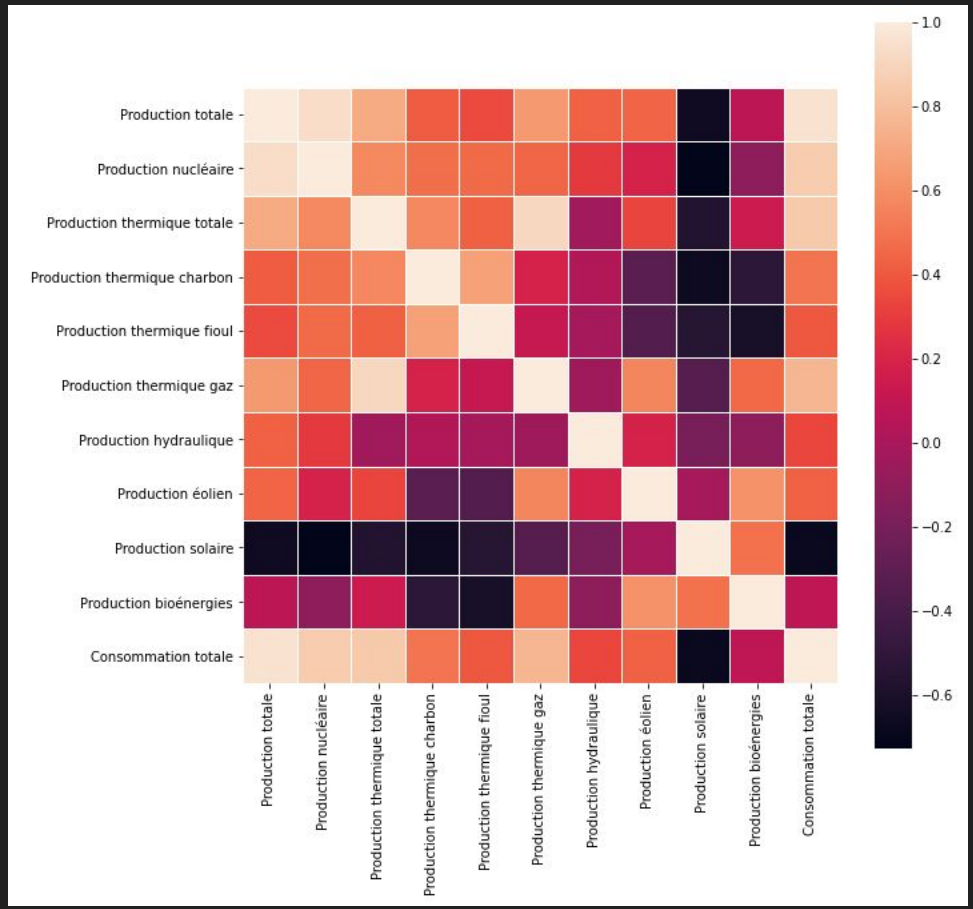
```
print((df.isna().sum()/df.shape[0]*100).round(2))
```

```
Qualité                                  0.00
Territoire                               0.00
Production totale                        0.00
Production nucléaire                     0.00
Production thermique totale              0.00
Production thermique charbon             0.00
Production thermique fioul               0.00
Production thermique gaz                 0.00
Production hydraulique                   0.00
Production éolien                        0.00
Production solaire                       0.00
Production bioénergies                   0.00
Consommation totale                      0.00
Solde exportateur                        0.00
Echanges export                          0.00
Echanges import                          0.00
Echanges avec le Royaume-Uni             0.00
Echanges avec l'Espagne                  0.00
Echanges avec l'Italie                   0.00
Echanges avec la Suisse                  0.00
Echanges avec l'Allemagne et la Belgique    10.92
dtype: float64
```

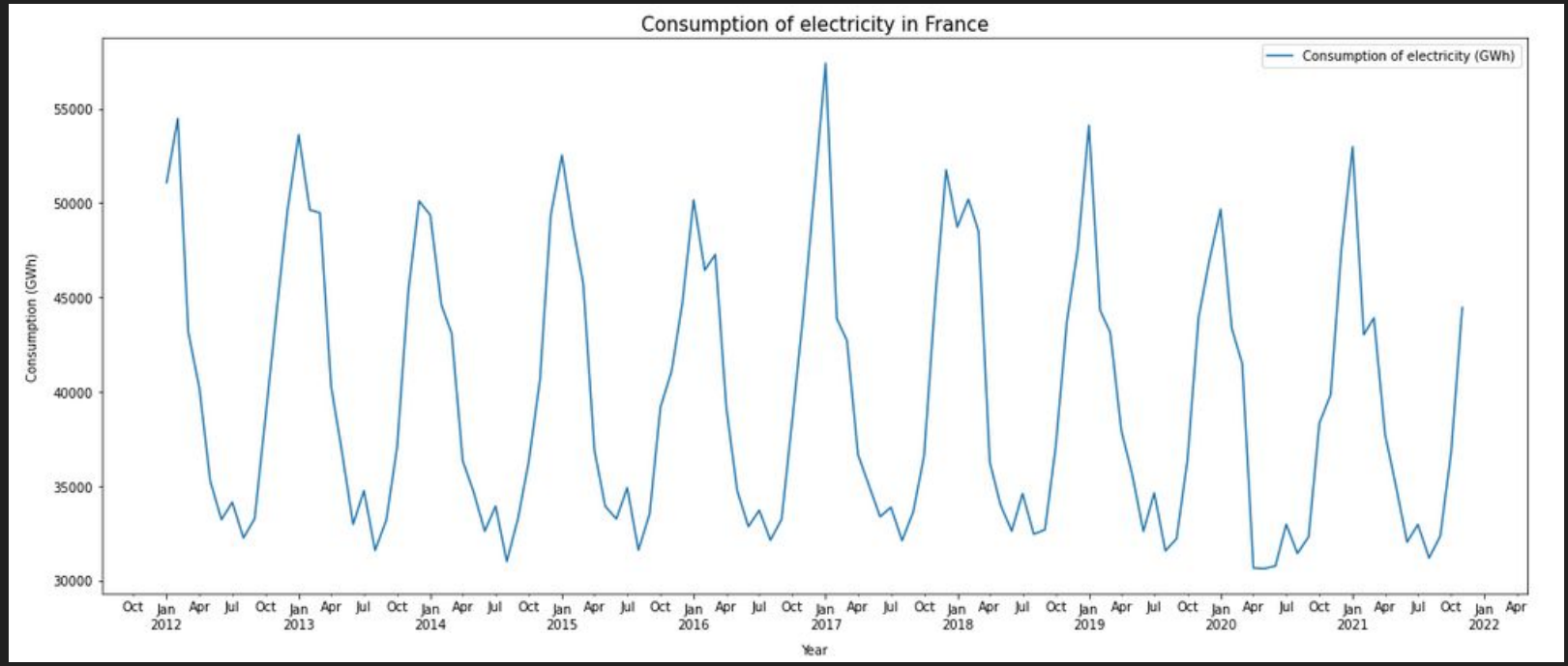
Nous isolons les données de la France et vérifions à nouveau les valeurs manquantes. Cette fois, nous n'avons aucune valeur manquante dans nos colonnes critiques (telles que la production et la consommation).

Relations entre les variables



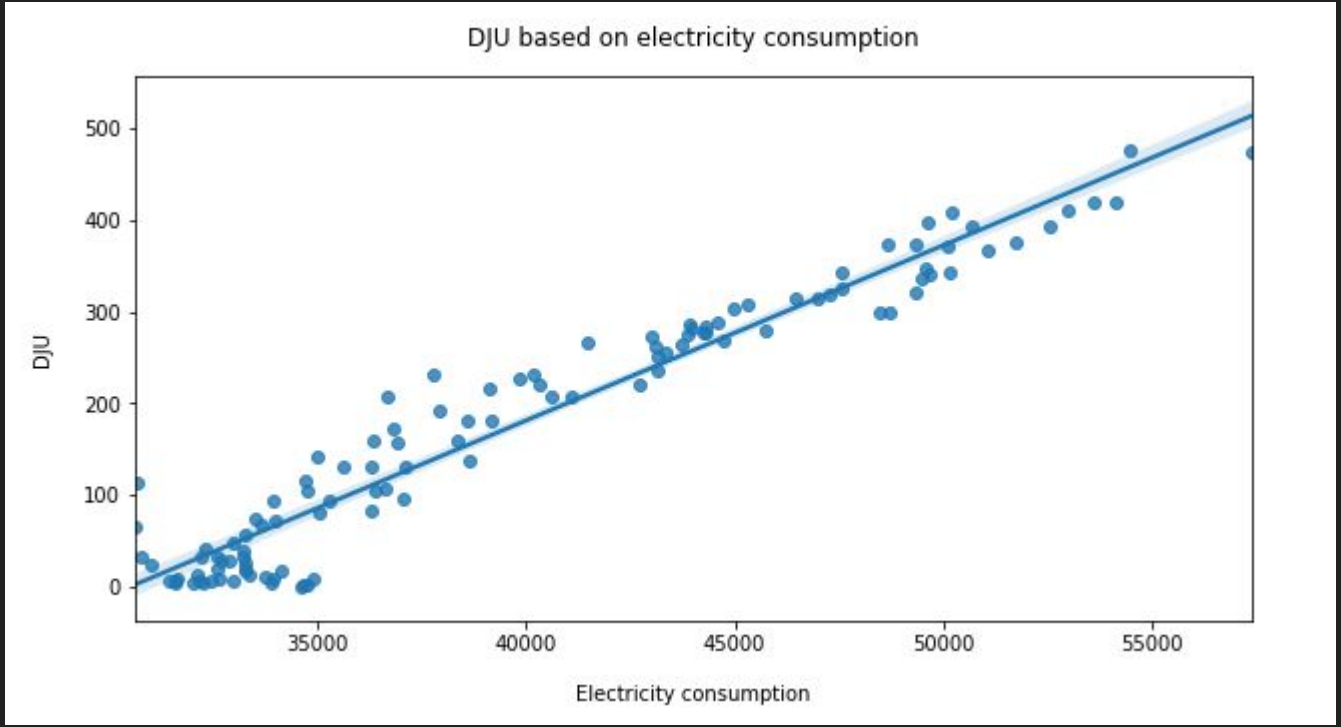


Consommation d'électricité en France



Lorsque nous traçons la consommation d'électricité de la France, nous pouvons voir son historique et des pics de saisonnalité évidents.

Corrigez les données de consommation mensuelles de l'effet température (dues au chauffage électrique) en utilisant une régression linéaire



Régression linéaire

```python
# Regression lineare avec statsmodels
# Linear regression with statsmodels
electr_total = df['Consommation totale']
DJU_var = df['DJU']

linreg = smf.ols('electr_total ~ DJU_var', data=df).fit()
print(linreg.summary())
```

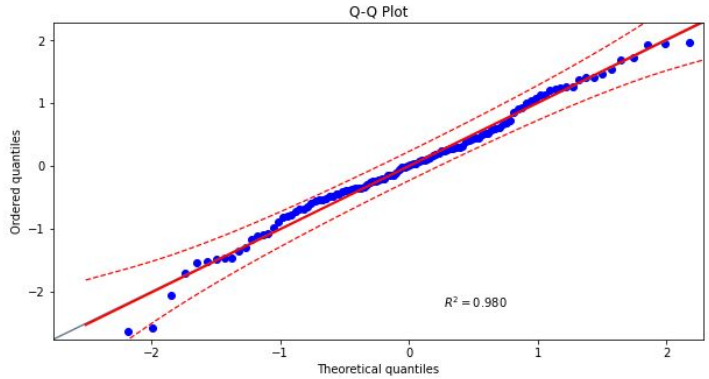
OLS Regression Results

Dep. Variable:	electr_total	R-squared:	0.937
Model:	OLS	Adj. R-squared:	0.936
Method:	Least Squares	F-statistic:	1654.
Date:	Tue, 29 Mar 2022	Prob (F-statistic):	6.57e-69
Time:	15:52:22	Log-Likelihood:	-1014.7
No. Observations:	114	AIC:	2033.
Df Residuals:	112	BIC:	2039.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	3.111e+04	271.567	114.544	0.000	3.06e+04	3.16e+04
DJU_var	48.9872	1.205	40.667	0.000	46.600	51.374

Omnibus:	5.080	Durbin-Watson:	1.531
Prob(Omnibus):	0.079	Jarque-Bera (JB):	4.589
Skew:	-0.387	Prob(JB):	0.101
Kurtosis:	3.606	Cond. No.	365.

```python
# tests statistiques pour vérifier la performance du modèle
# statistical tests to check performance of the model

pg.qqplot(linreg.resid, figsize=(10,5))

plt.savefig(fname=f'LinRegression_QQplot.jpg', dpi=100, format='jpg', orien
plt.show();
```

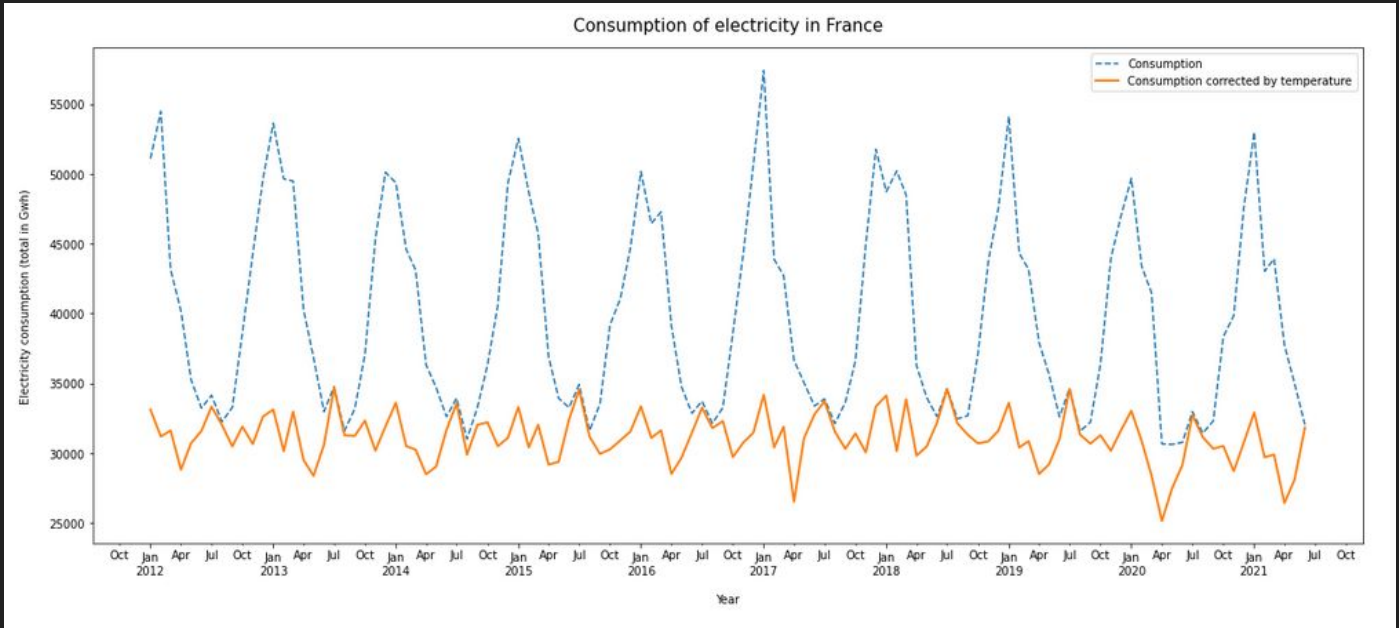



```python
# https://pingouin-stats.org/generated/pingouin.

pg.normality(linreg.resid, method='normaltest')
```

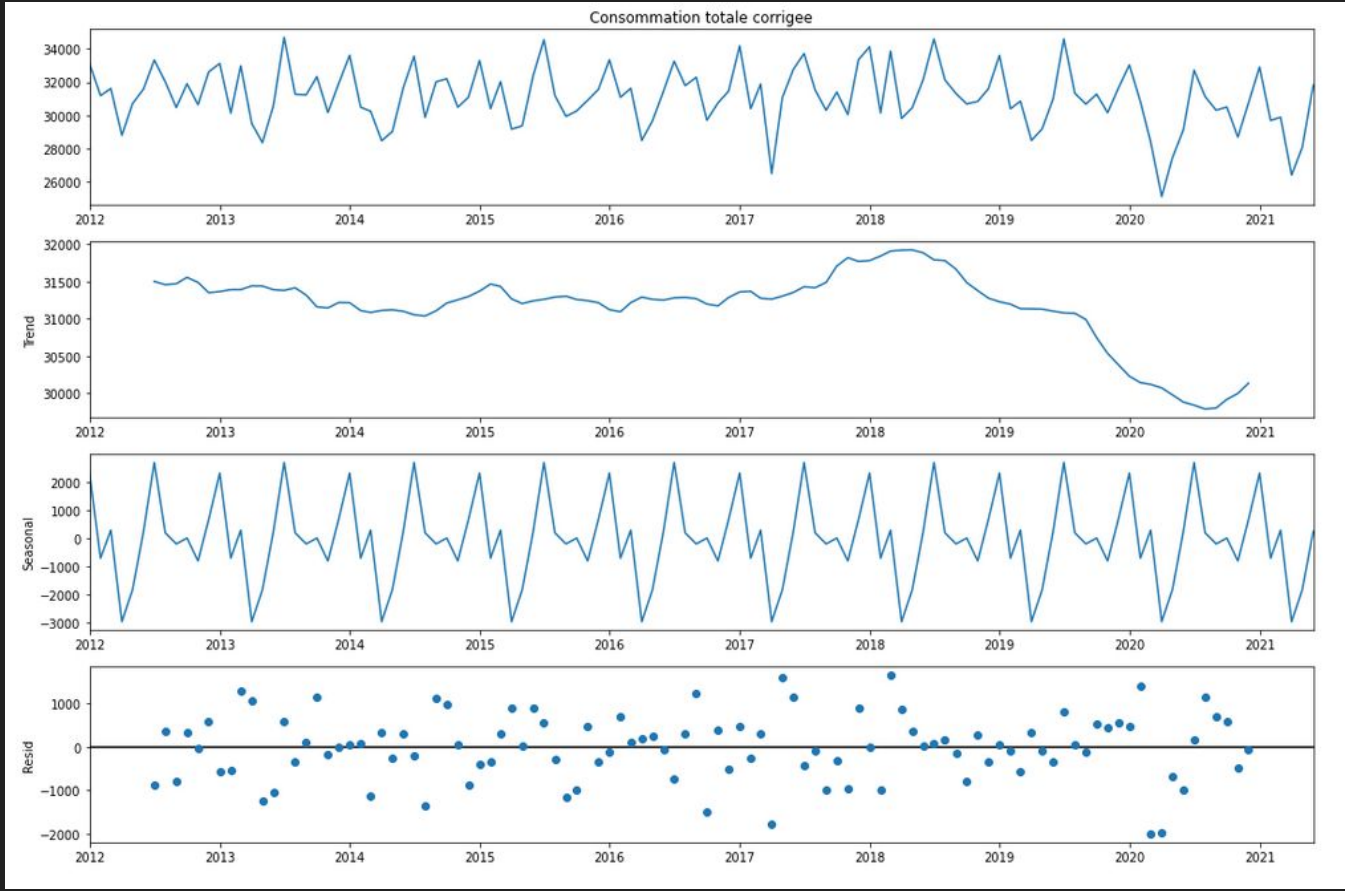
	W	pval	normal
0	5.079996	0.078867	True

L'évolution de l'indice de Gini

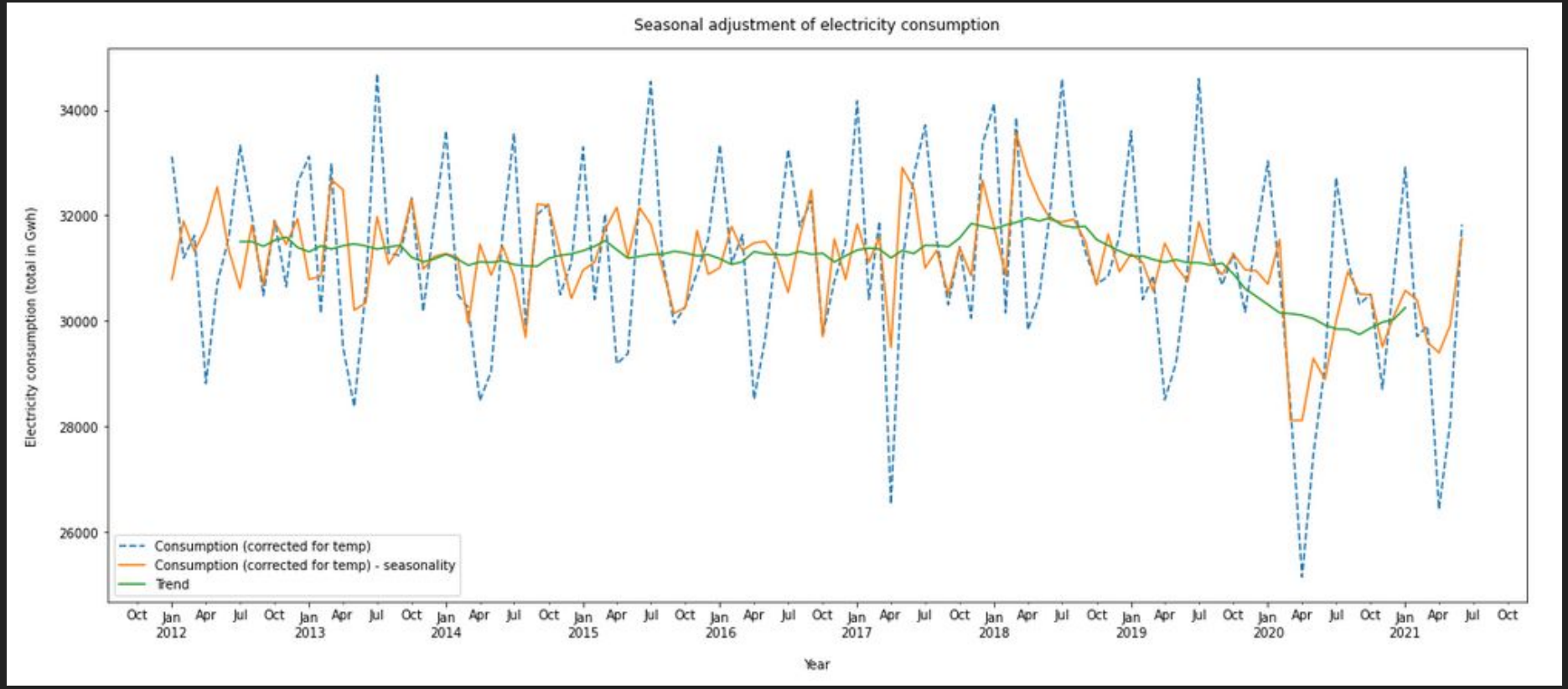
Le graphique QQ donne une représentation visuelle que les résidus suivent de près la distribution normale. Le test de normalité le confirme avec une valeur p de 7% (nous acceptons donc H0 que les données sont normalement distribuées). Mais le test de variance renvoie une valeur p très élevée de ~80%, ce qui signifie que nous rejetons H0 selon lequel les variances sont constantes (elles ne le sont pas selon ce test).



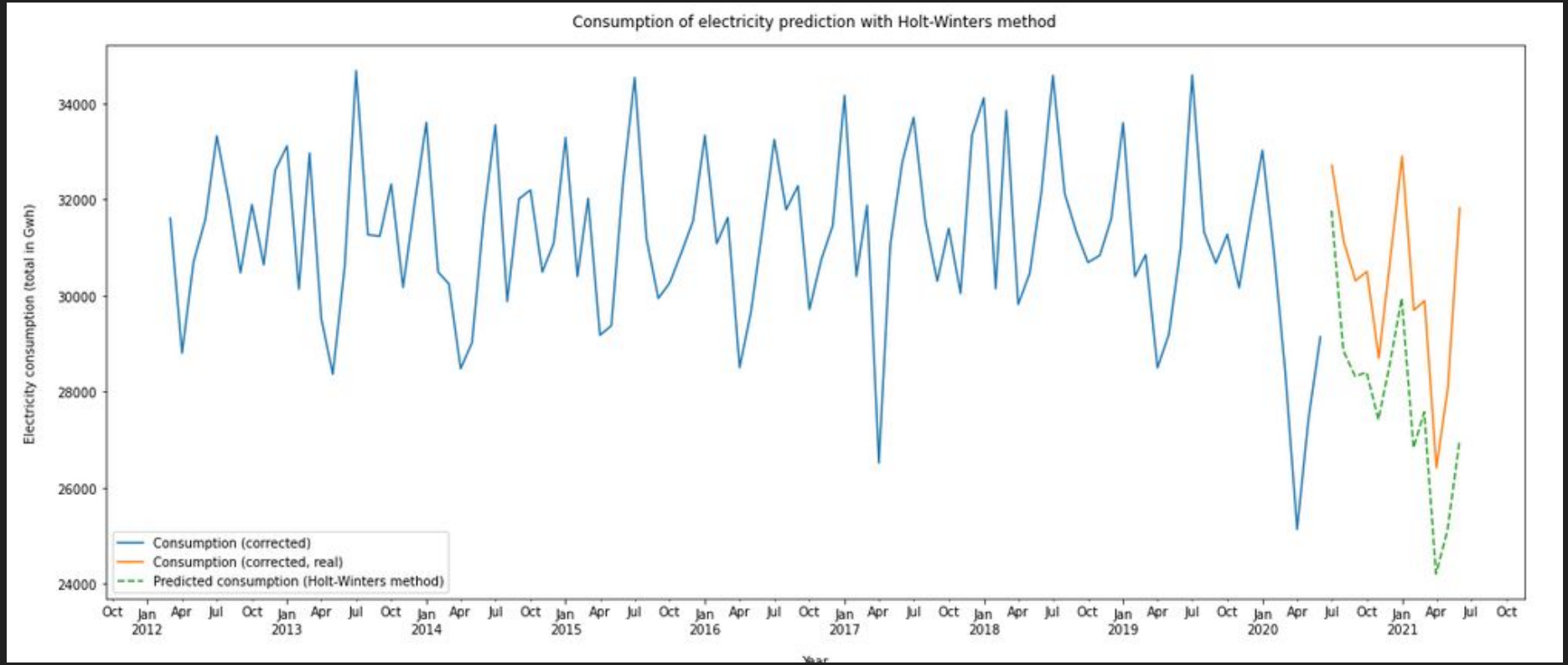
Désaisonnalisation de la consommation



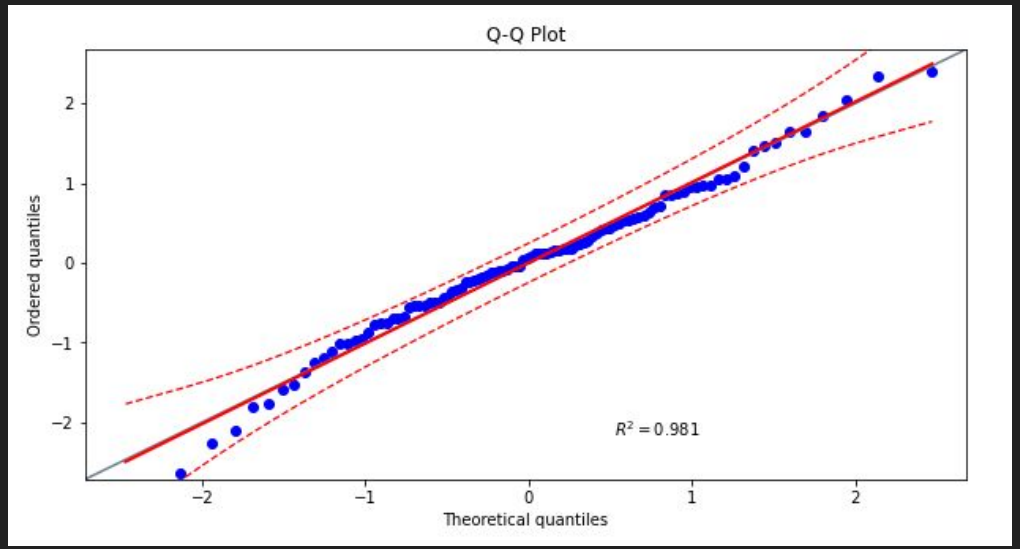
Désaisonnalisation de la consommation



Modèle de prédiction, Holt Winters



Modèle de prédiction, Holt Winters



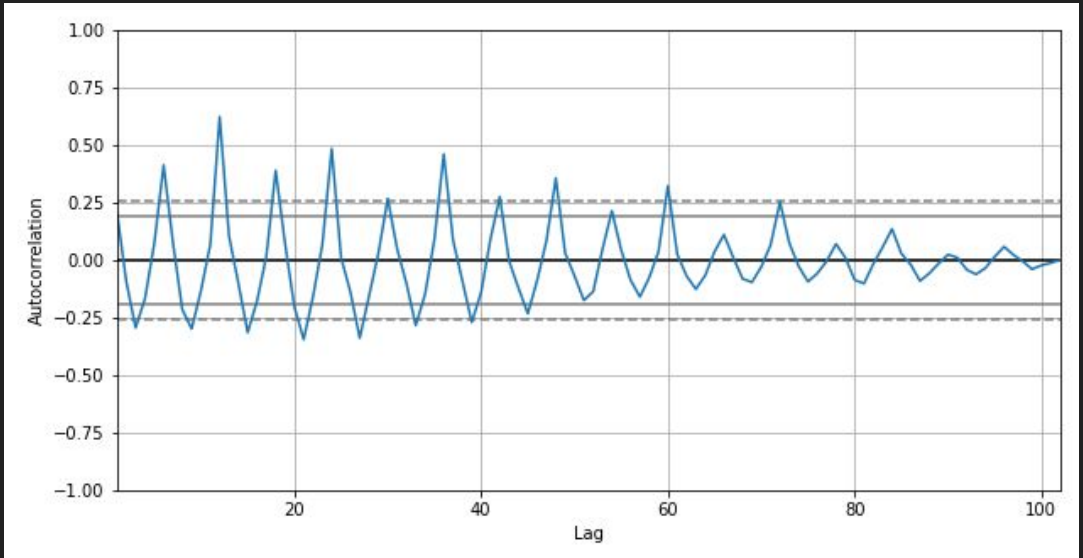

```
3 pg.normality(hw.resid, method='normaltest')
```

	W	pval	normal
0	4.859223	0.088071	True

```
RMSE =  2591.62
MAPE =  7.98
```

Les tests confirment que nos données sont normalement distribuées. Le graphique QQ donne une représentation visuelle, et le test de normalité donne une valeur p de 8% qui confirme H0 (la population est normalement distribuée).

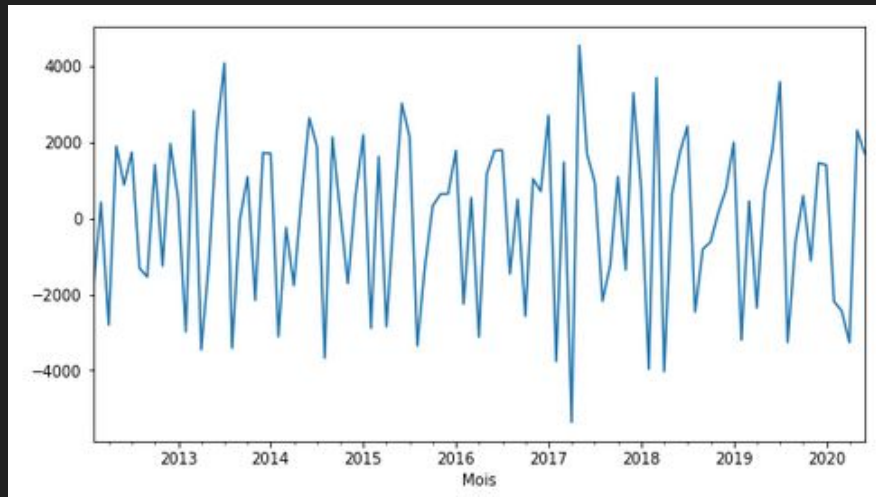
Modèle de prédiction, SARIMA

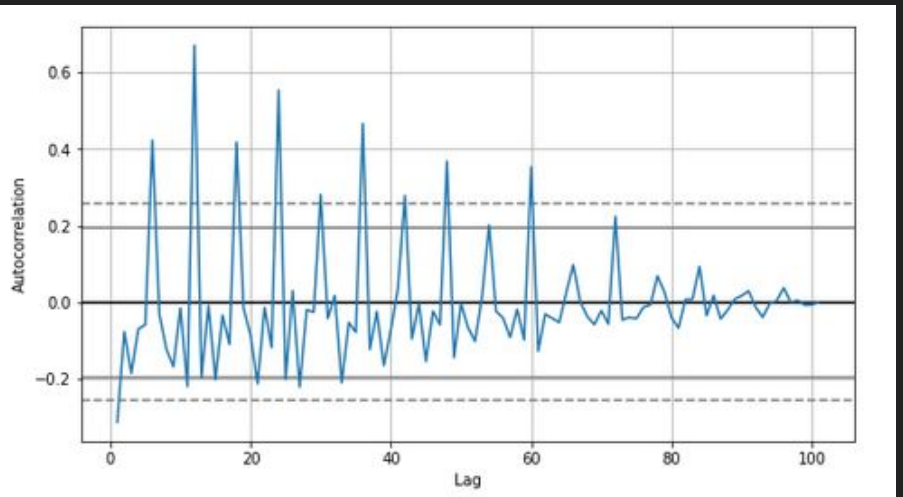


L'autocorrélogramme montre que notre série chronologique n'est pas stationnaire.

Nous devons atteindre la stationnarité afin d'être en mesure de modéliser les données futures.

Modèle de prédiction, SARIMA

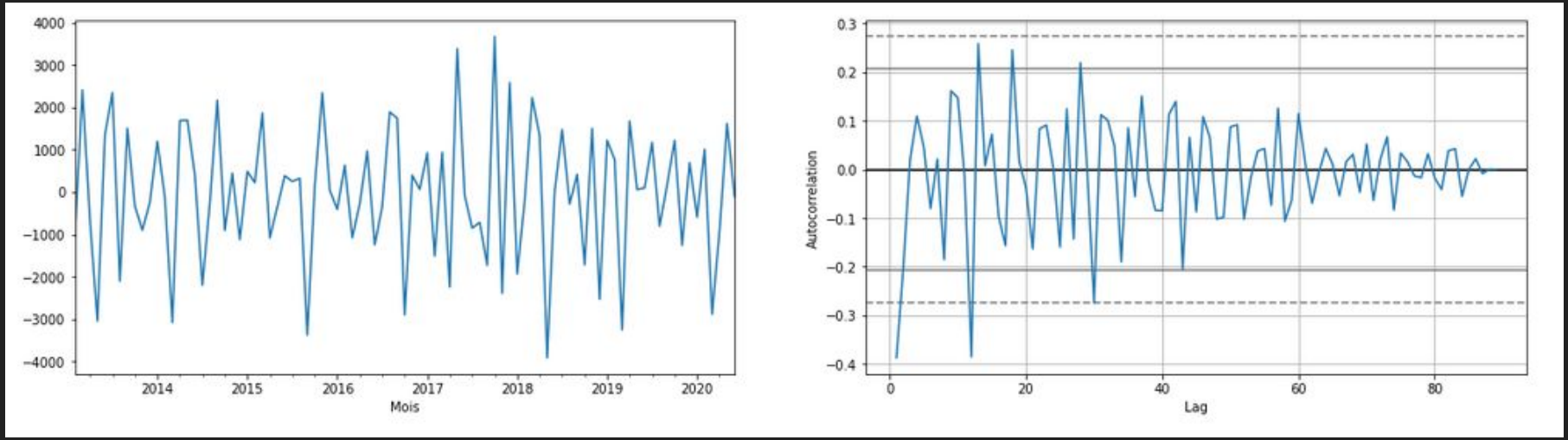




L'autocorrélation tend à nouveau vers 0, mais nous pouvons clairement voir des pics saisonniers.

Nous pouvons essayer la différenciation de (I-B12).

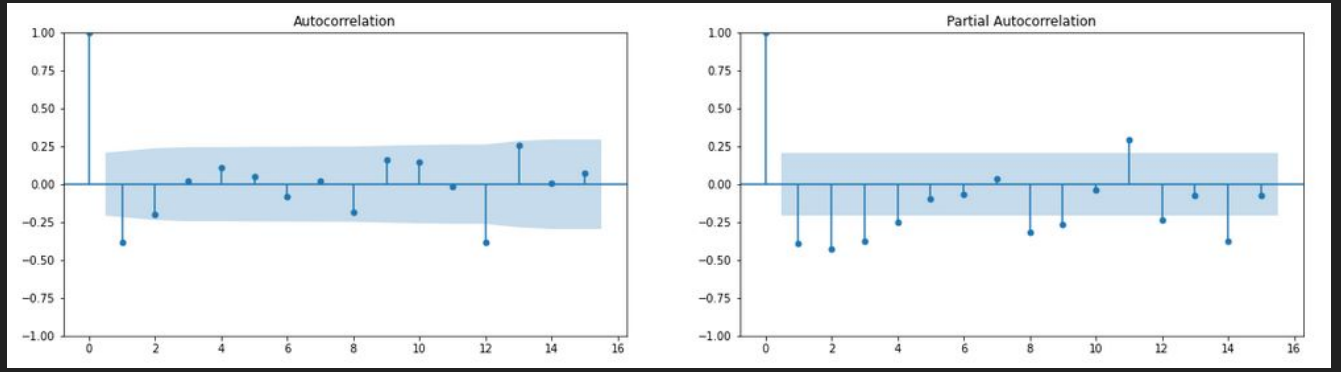
Modèle de prédiction, SARIMA



Le résultat est meilleur ici. Les pics existent toujours, mais les valeurs diminuent plus rapidement et semblent maintenant plus stationnaires.

Répète le test pour confirmer et la valeur p du test de Dickey Fuller est inférieure à 5%, ce qui signifie que la série temporelle est stationnaire.

Modèle de prédiction, SARIMA



SARIMAX Results

Dep. Variable:	Consommation totale corrigee	No. Observations:	102
Model:	SARIMAX(1, 1, 1)x(0, 1, 1, 12)	Log Likelihood	-774.698
Date:	Tue, 29 Mar 2022	AIC	1557.396
Time:	15:52:32	BIC	1567.350
Sample:	01-01-2012	HQIC	1561.408
	- 06-01-2020		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.3198	0.252	1.272	0.204	-0.173	0.813
ma.L1	-0.4825	0.251	-1.926	0.054	-0.974	0.009
ma.S.L12	-0.2448	0.047	-5.199	0.000	-0.337	-0.153
sigma2	1.934e+06	3.03e+05	6.374	0.000	1.34e+06	2.53e+06

Ljung-Box (L1) (Q):	8.09	Jarque-Bera (JB):	0.76
Prob(Q):	0.00	Prob(JB):	0.68
Heteroskedasticity (H):	0.73	Skew:	-0.23
Prob(H) (two-sided):	0.40	Kurtosis:	2.95

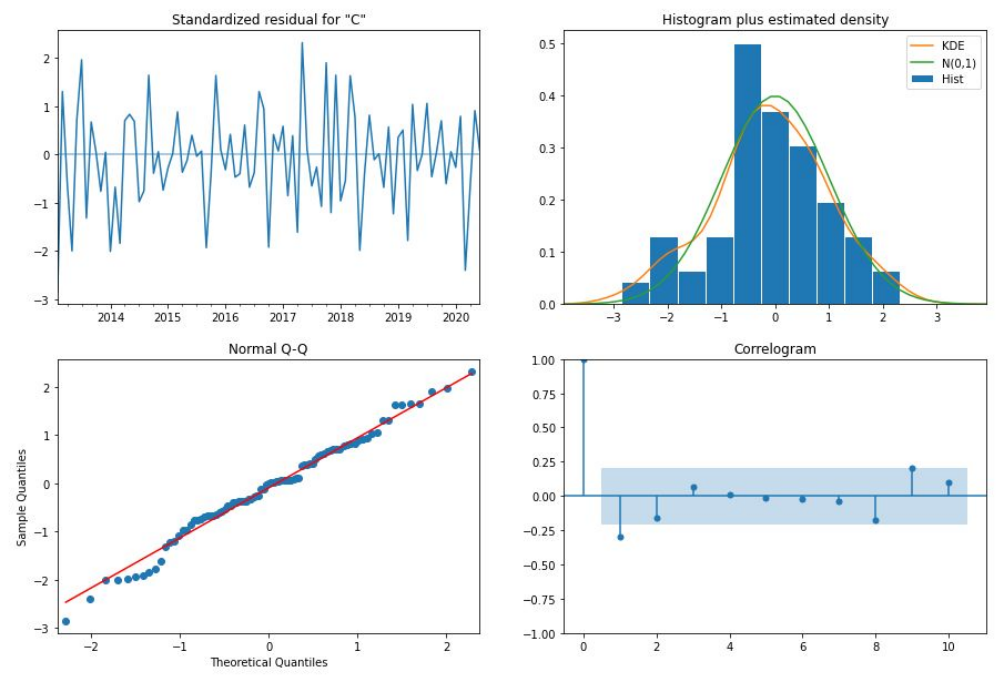
SARIMAX Results

Dep. Variable:	Consommation totale corrigee	No. Observations:	102
Model:	SARIMAX(0, 1, 1)x(0, 1, 1, 12)	Log Likelihood	-775.541
Date:	Tue, 29 Mar 2022	AIC	1557.082
Time:	15:52:32	BIC	1564.548
Sample:	01-01-2012	HQIC	1560.092
	- 06-01-2020		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.1603	0.029	-5.481	0.000	-0.218	-0.103
ma.S.L12	-0.2576	0.046	-5.569	0.000	-0.348	-0.167
sigma2	1.98e+06	3.13e+05	6.328	0.000	1.37e+06	2.59e+06

Ljung-Box (L1) (Q):	8.10	Jarque-Bera (JB):	0.57
Prob(Q):	0.00	Prob(JB):	0.75
Heteroskedasticity (H):	0.71	Skew:	-0.19
Prob(H) (two-sided):	0.36	Kurtosis:	2.95

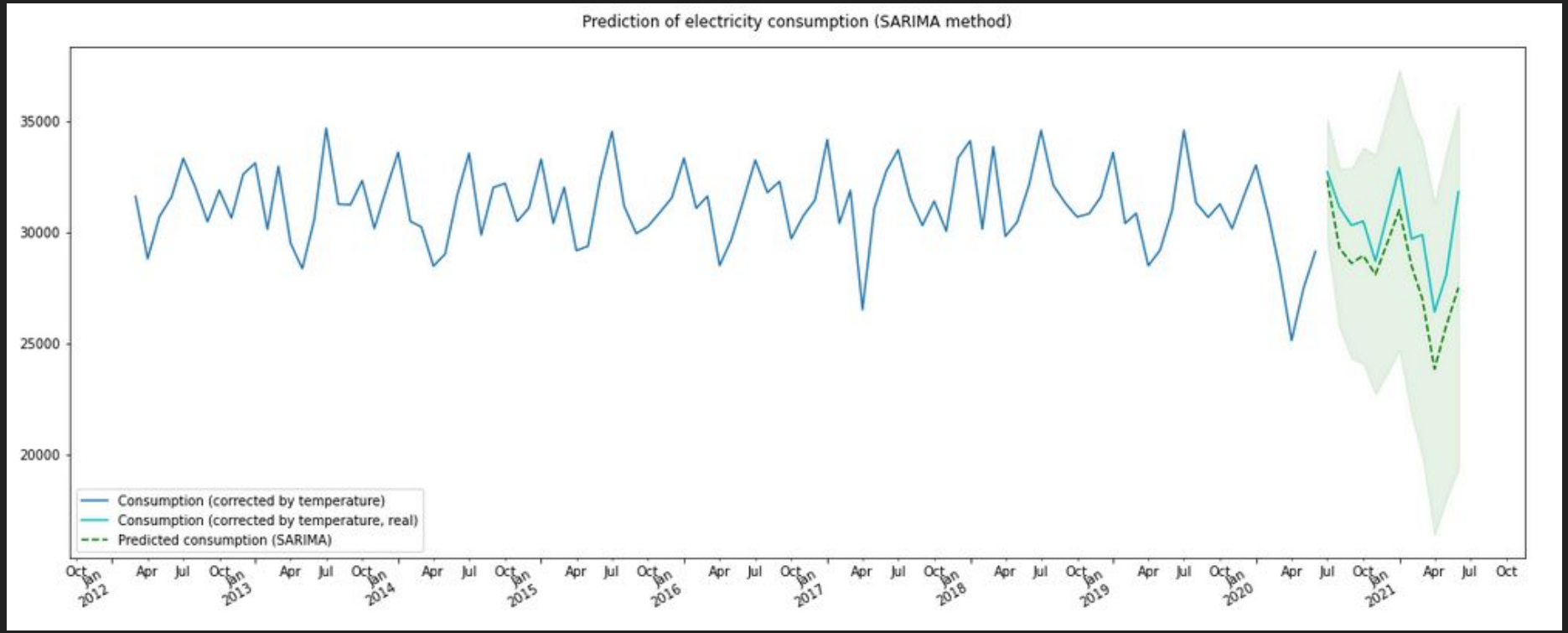
Modèle de prédiction, SARIMA



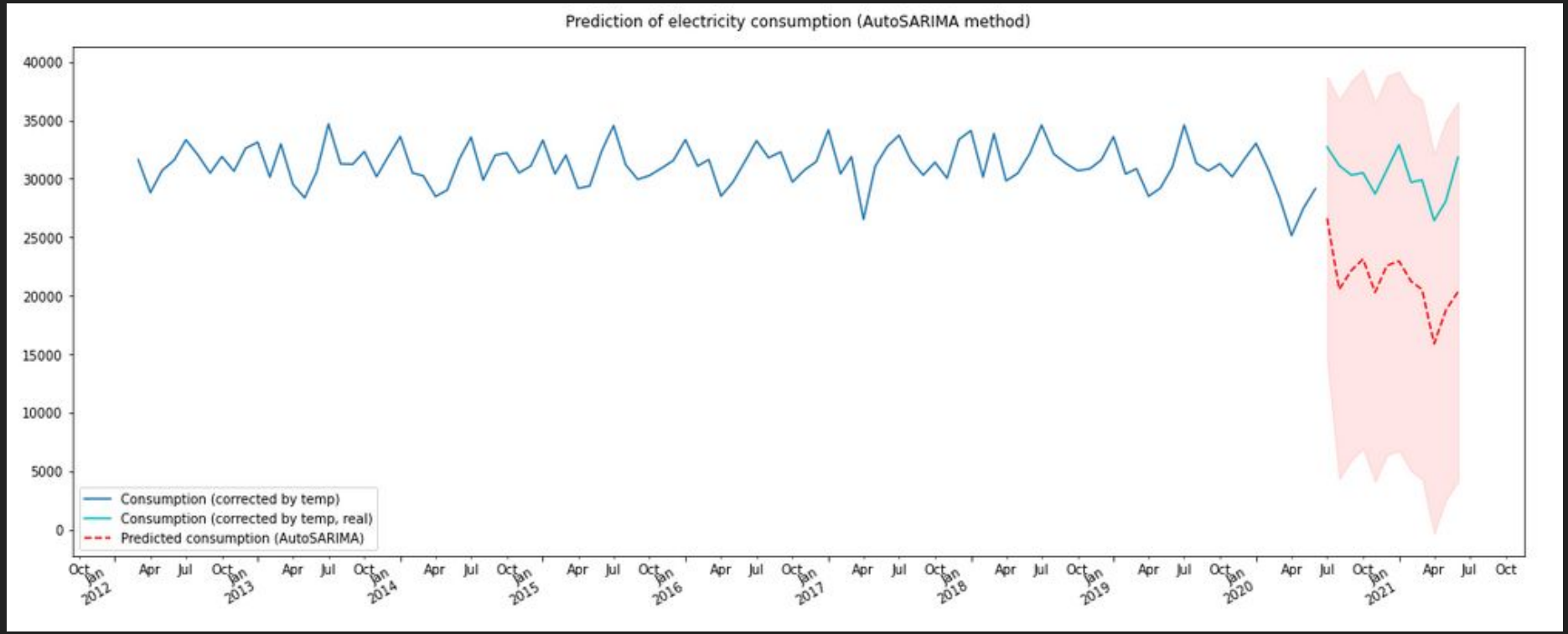
Les diagnostics SARIMA aident (un bon result est...):

- Résidus normalisés - pas de modèle dans les résidus.
- Histogramme plus densité estimée - ils doivent être proches de la distribution normale.
- QQ normal - aussi proche de la ligne de la distribution normale que possible.-
- Corrélogramme - 95 % des points de corrélation qui sont > 0 ne doivent pas être significatifs.

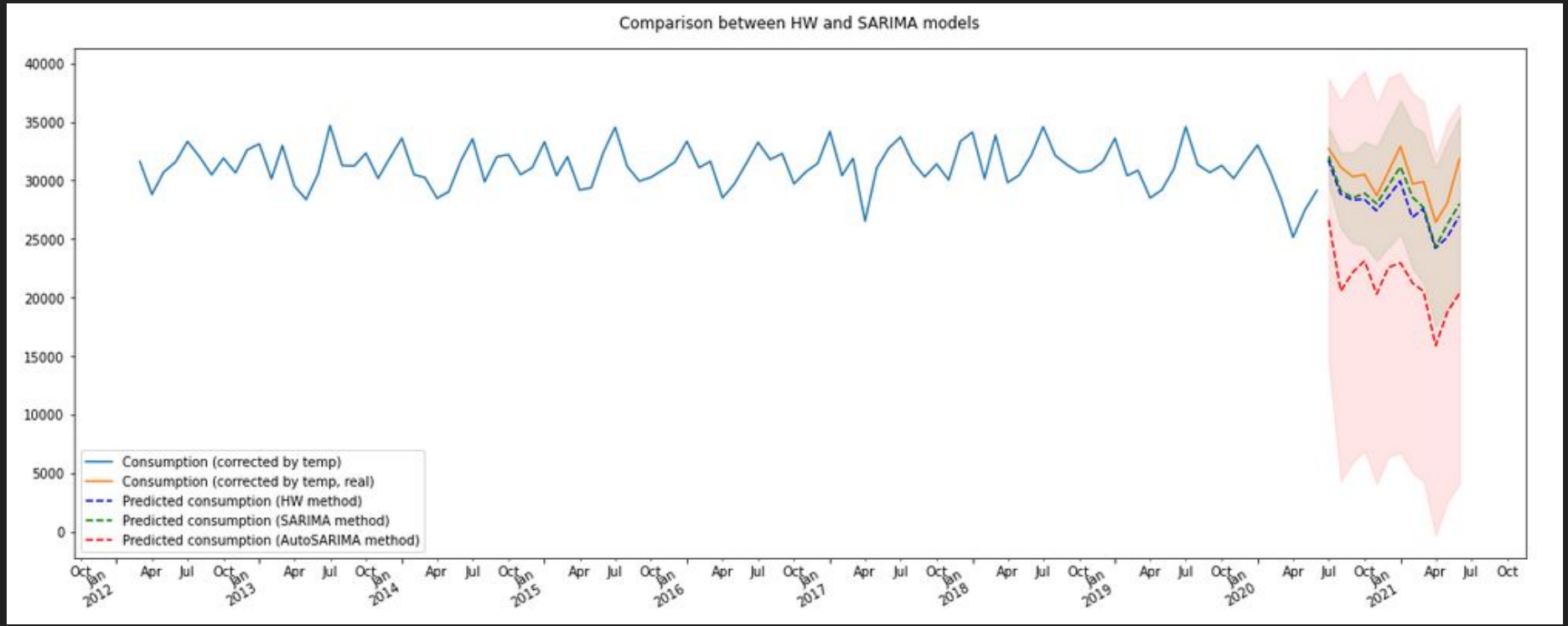
Prédiction avec SARIMA



Modèle de prédiction, AutoARIMA



Comparaison des modèles



Conclusions générales après l'analyse effectuée

L'utilisation des autocorrélogrammes pour déterminer les modèles SARIMA n'est pas une règle absolue. Elle peut toujours dépendre de la situation. C'est pourquoi les autocorrélogrammes doivent servir d'indicateur utile, en collaboration avec la blancheur des résidus et les degrés de signification. De nombreux modèles peuvent être considérés comme bons après validation de ces paramètres. Dans ce cas, nous pouvons utiliser l'AIC, le BIC, le HQIC donnés par la sortie de SARIMA.
Au vu de notre analyse, nous pouvons dire que le modèle SARIMA donne les meilleurs résultats. Tant sur le plan visuel si nous le comparons aux données réelles, que sur le plan statistique si nous le comparons aux autres modèles.
Si nous devions mettre en œuvre un modèle spécifique, ce serait le modèle SARIMA (ligne pointillée verte).

The use of autocorrelograms to determine SARIMA models is not a strong rule. It can always depend on the situation. Because of this autocorrelograms should serve as a helpful indicator, working together with whiteness of residuals and degrees of significance. Many models can be seen as good after validation of those parameters. In this case we can use AIC, BIC, HQIC given by SARIMA output.
Given our analysis, we can say that SARIMA model yields the best results. Both visually if we compare to the real data, and statistically when we compare metrics with other models.
If we were to put one specific model to work, it would be SARIMA (green dotted line).