OPENCLASSROOMS

Mission 9

Effectuez une prédiction de revenus

Mentor : Claire Della Vedova

Laurent Gourdon – Parcours Data Analyst

Le contexte

Enercoop est une société coopérative spécialisée dans les énergies renouvelables qui s'est développée grâce à la libéralisation du marché de l'électricité en France

Les contraintes

- Une grande part des énergies renouvelables est intermittente.
- La demande en électricité des utilisateurs varie au cours du temps, dépend de la météo (température, ensoleillement, etc.) et de la localisation géographique

enercoop
L'énergie militante

Afin de mettre en équation l'offre d'ENERCOP et la demande, nous allons réaliser une étude préliminaire à partir d'un historique de la consommation en France et chercher un modèle prédictif adéquat à partir de plusieurs méthodes utilisées pour les séries temporelles

Agenda

Data wrangling + analyse descriptive
- Consommation électrique France
- Données journalières unifiées Paris

Mission 1
- Regression linéaire
- Diagnostics
- Données normalisées

Mission 2
- Décomposition
- Désaisonnalisation (MA)
- Représentations graphiques et analyse

Mission 3
- Prévision Holt-Winters
- Prévisions SARIMA
- Choix du modèle

Conclusion

Les données

- Dans le fichier eCO2mix, la variable “Territoire” contient des données à l’échelle régionale et nationale.
- Les régions offrent 6 années complètes (2013-2018) alors que le niveau national offre 9 années complètes (2010-2018) de consommation.
- Les années 2010-2013 sont des données consolidées alors que 2014-202018 sont des données définitives
- Nous choisissons Territoire = France pour cette étude préléminaire afin de disposer d’un maximum d’années d’observation

Nettoyage des données

- Territoire = France
- Qualité = 'Données définitives' 'Données consolidées’
- Data = integer
- Index = time serie
- Period = mensuelle
- Années = 2010-2018

1.1.1 Selecting data for Territoire = France

```
# sélection de 'France'
cvl=data[data['Territoire']=='France'].reset_index(drop=True)
# Rename
cvl.rename(columns={'Mois': 'period','Consommation totale': 'conso'},
           inplace = True)
print('Qualité :',cvl['Qualité'].unique())
```

```
Qualité : ['Données définitives' 'Données consolidées']
```

```
# drop cols
cvl.drop(['Qualité', 'Territoire'], axis=1, inplace=True)
# supress incorrect/useless values
cvl = cvl[cvl['conso'] != 0]
cvl = cvl[cvl['period'] != '0000-00']
cvl = cvl[cvl.period < '2019-01']
# dtypes convertion
cvl = cvl.astype({"conso":int})
```

1.1.2 Setting time-serie

```
# time-serie
cvl.period=pd.to_datetime(cvl.period, format="%Y-%m-%d", dayfirst=False)
# set index = period
cvl.set_index('period', inplace=True)
# set période = month
cvl=cvl.to_period('M')
cvl.head(2)
```

	conso
period	
2010-01	56342
2010-02	48698

CEGIBAT

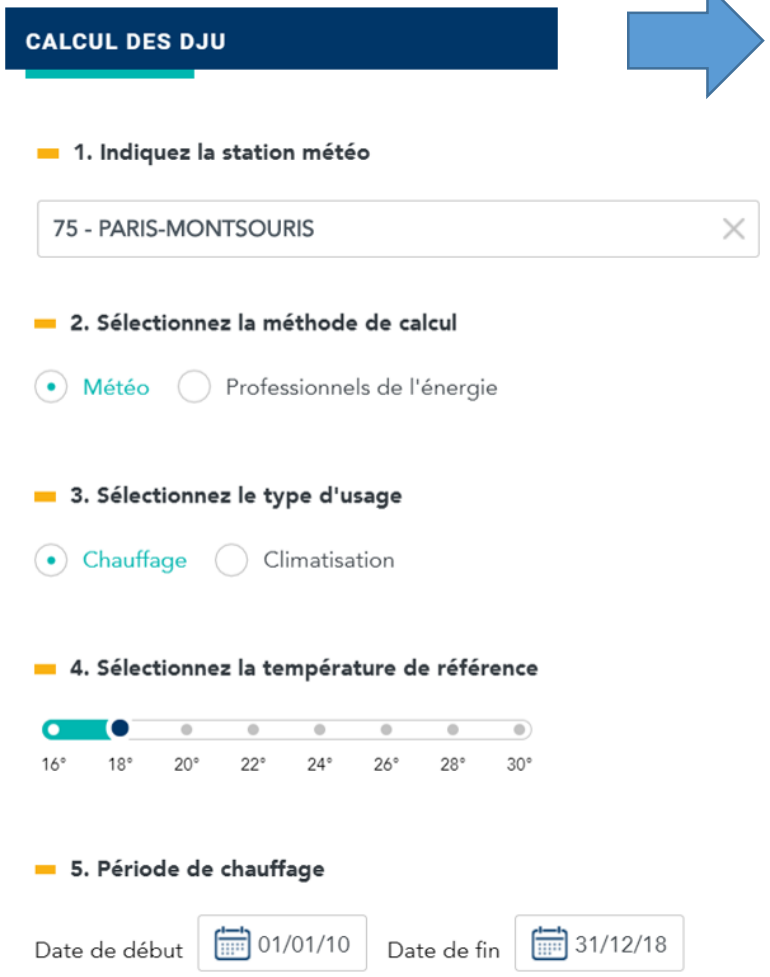
L'expertise efficacité énergétique de GRDF

La normalisation

- L'outil, réalisé en partenariat avec Météo France, permet de calculer les degrés jour (DJ ou DJU) chauffage ou climatisation sur une période, une station météo et un seuil de température donnés
- Le **degré jour** est une valeur représentative de l'écart entre la température d'une journée donnée et un seuil de température préétabli (18 °C dans le cas des DJU ou Degré Jour Unifié). Sommés sur une période, ils permettent de calculer les besoins de chauffage et de climatisation d'un bâtiment.

DJU 18ºC Paris

CEGIBAT
L'expertise efficacité énergétique de GRDF

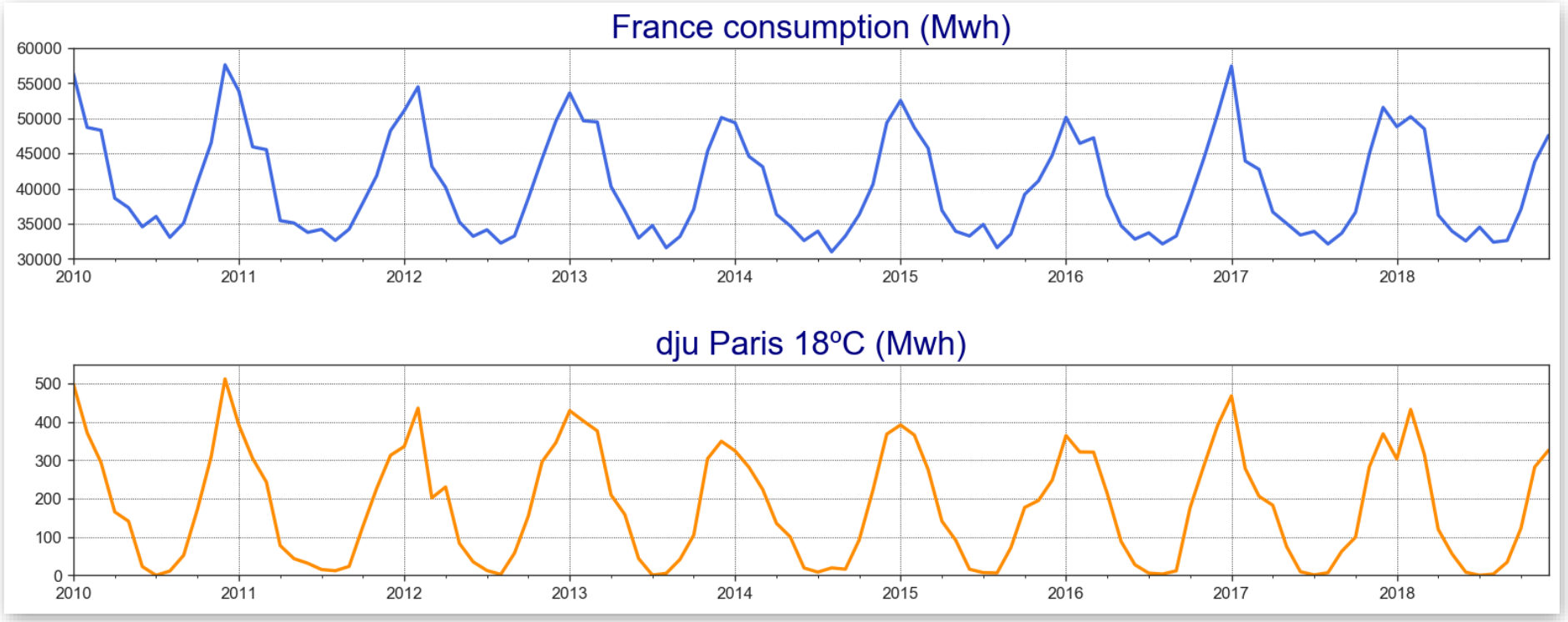


	Jan	Fév	Mar	Avr	Mai	Jun	Jui	Aoû	Sep	Oct	Nov	Déc	Total
2018	303	433	314	120	56	8	0	3	34	122	283	326	2 002
2017	468	278	206	183	75	9	1	7	63	99	283	369	2 041
2016	364	322	321	212	88	28	6	3	12	176	286	391	2 207
2015	392	366	276	141	92	16	7	6	72	177	195	248	1 986
2014	324	282	224	136	100	19	8	19	16	92	223	368	1 812
2013	429	402	377	210	158	44	1	5	42	105	304	350	2 425
2012	336	436	202	230	83	35	12	2	58	155	296	346	2 192
2011	392	305	243	78	43	31	15	12	23	128	227	313	1 809
2010	499	371	295	165	141	23	0	11	52	172	310	512	2 551

dju

period	
2018-01	303.4
2018-02	432.6
2018-03	314.3

Index = time serie
Period = mensuelle
Années = 2010-2018



Données consolidées

Forte corrélation entre les deux jeux de données

Régression linéaire

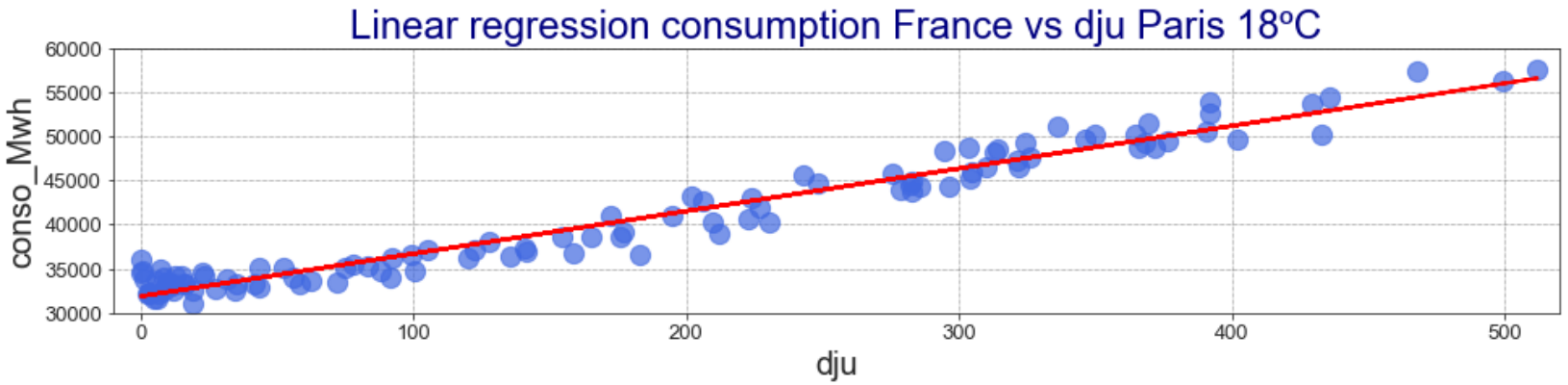
$$lr = \alpha + \beta.dju + \epsilon$$

La regression linéaire conso vs dju nous permet d'estimer l'augmentation de la consommation pour 1 dju

OLS Regression Results

Dep. Variable:	conso_Mwh	R-squared:	0.956
Model:	OLS	Adj. R-squared:	0.955
Method:	Least Squares	F-statistic:	2282.
Date:	Sat, 11 Jan 2020	Prob (F-statistic):	1.60e-73
Time:	11:22:47	Log-Likelihood:	-944.71
No. Observations:	108	AIC:	1893.
Df Residuals:	106	BIC:	1899.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
dju	48.2471	1.010	47.768	0.000	46.245	50.250
const	3.188e+04	231.396	137.753	0.000	3.14e+04	3.23e+04



- **R-squared** = 0,956 confirme la correlation entre les 2 series
- **α = 31880**
- **β = 48.25**

Diagnostics

Pearson's Correlation Coefficient
Tests if a data has a Gaussian distribution

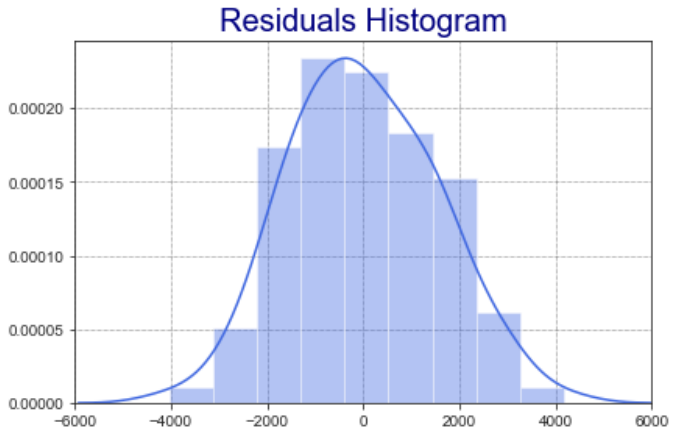
- H0: the two samples are independent
- H1: dependency between the samples

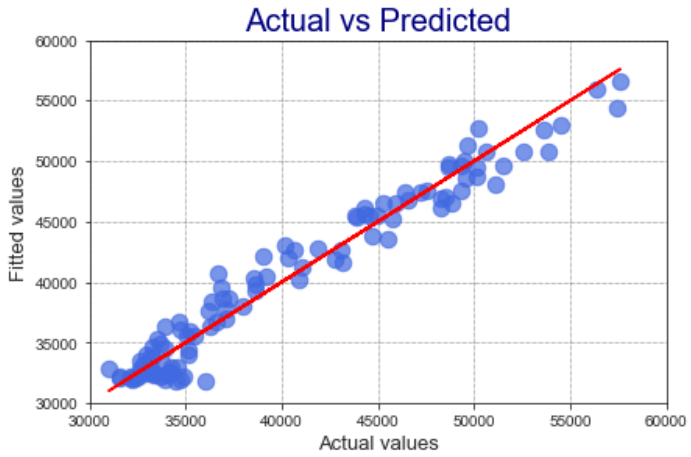
stat=0.978, p=0.000 - *probably dependent*

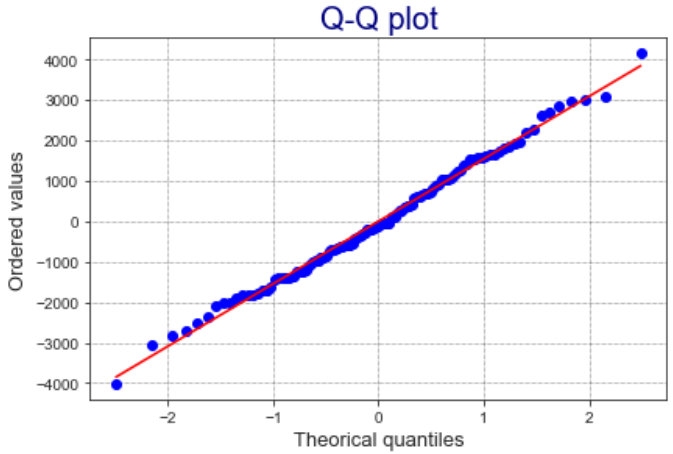
Normality: Shapiro-Wilk Test
Observations are independent and identically distributed (iid)

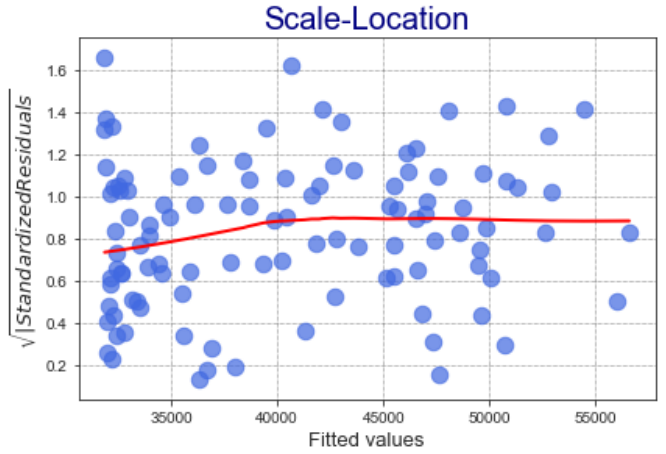
- H0: has a Gaussian distribution
- H1: does not have a Gaussian distribution

stat=0.995, p=0.950 - *probably Gaussian*









Normalisation de la consommation à 18ºC

conso_cor = conso - β.dju = α + ϵ

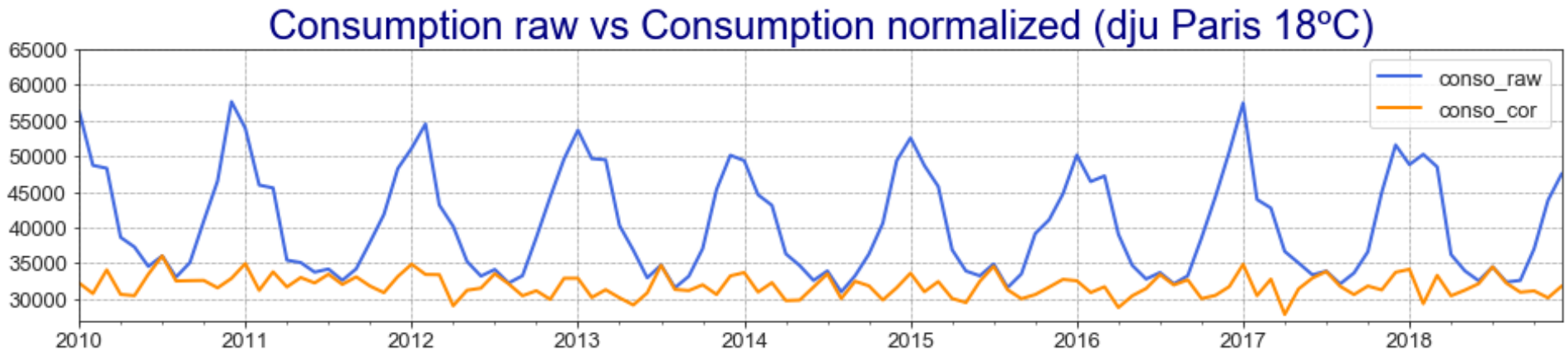
Conso électrique = chauffage + autre

Conso électrique normalisée 18ºC
= Conso électrique – chauffage = autre

```
# Correction
β = lin_reg.params['dju']
cvl2=pd.DataFrame(lr["conso_Mwh"]- lr['dju']*β)

# Rename
cvl2.rename(columns={0:'conso_cor'},  inplace = True)
cvl2.head(2)
```

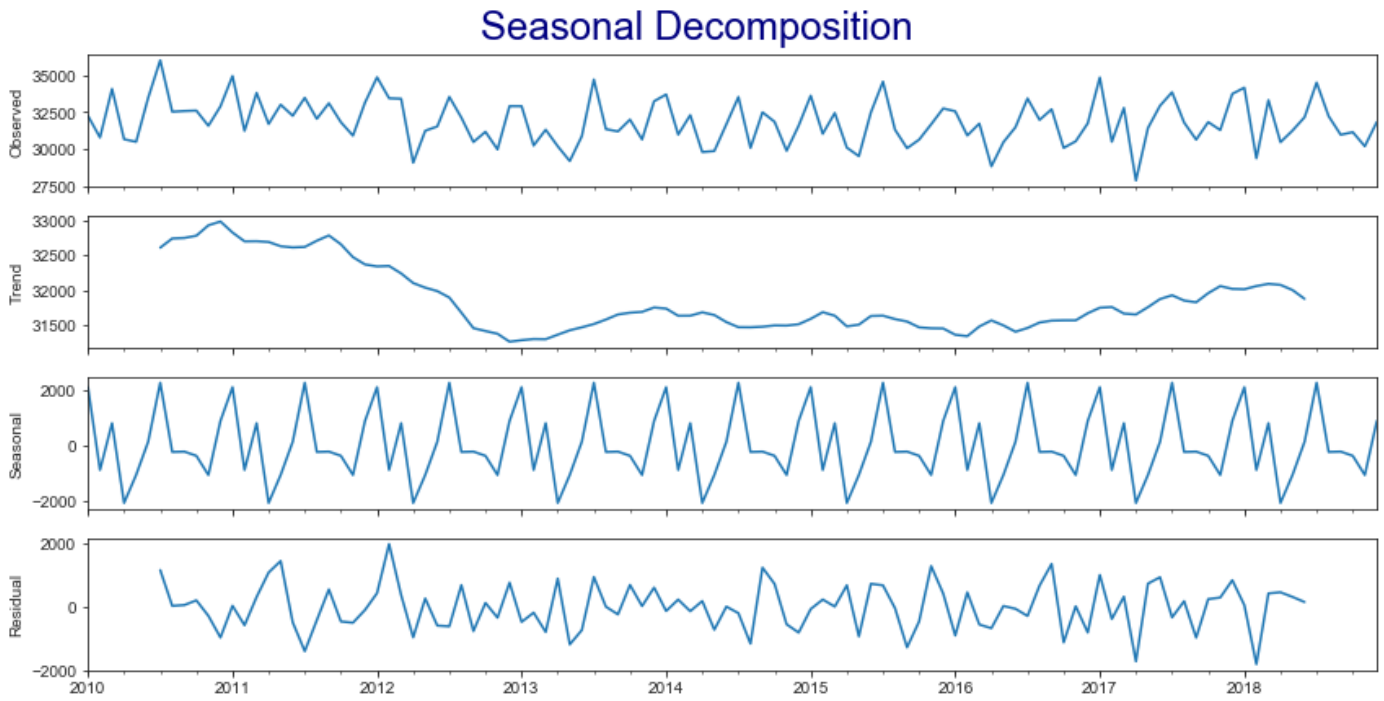
	conso_cor
period	
2010-01	32257.1
2010-02	30779.0



Désaisonalisation par moving average

Une série temporelle a 3 composants : Trend, Seasonalité, Résidus

- Objectif de la désaisonnalisation: Absorber la seasonalité, conserver la trend et minimizer la variance des résidus
- Considérant la courbe “Observed” nous choisirons le modèle additif (intervalle constant autour de la tendance), plutôt que multiplicative (intervalle proportionnel à la tendance)

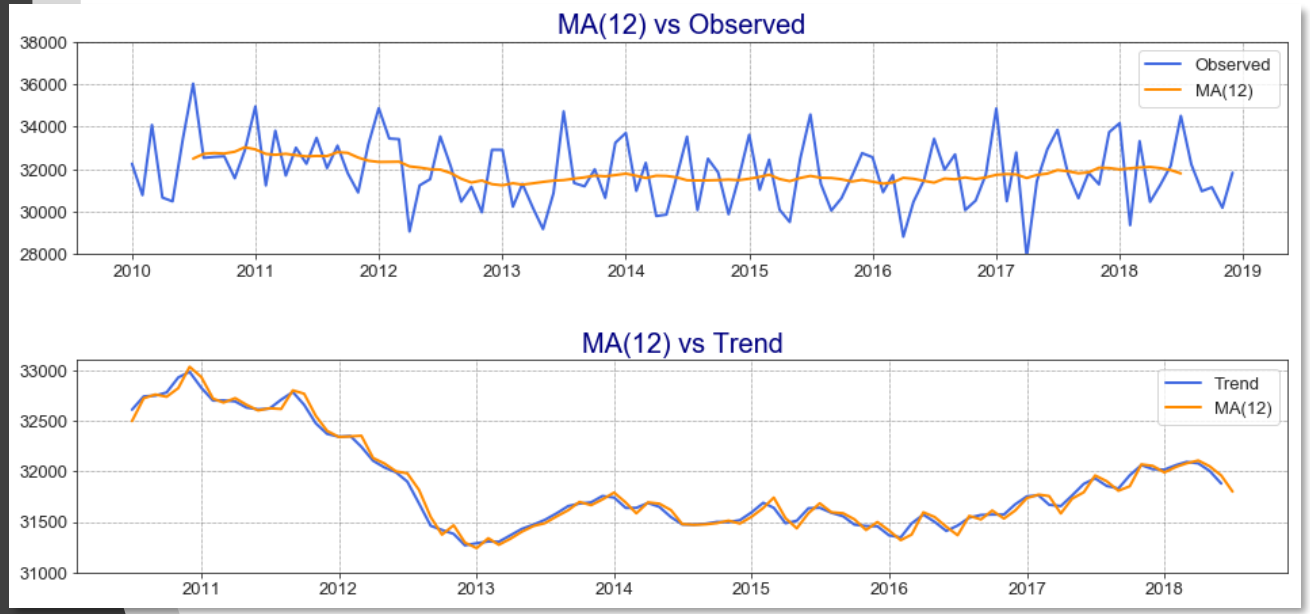


- **Observed** a un intervalle constant autour de la tendance => Modèle additif
- **Trend** est non-linéaire (plat, ascendant ou descendant)
- **Seasonal** montre une périodicité de 6 ou 12 mois
- **Residual** semble être du bruit

Désaisonalisation par moving average

- La désaisonnalisation MA(12) est efficace, très proche de la tendance obtenue pour la decomposition de la série temporelle
- L'influence des résidus est minimale

```
# Rolling average
ra=conso.conso_cor.rolling(window=12, center=True).mean()
```



Stationarité

Augmented Dickey-Fuller Unit Root Test
Tests whether a time series has a unit root, e.g. has a trend or more generally is autoregressive.

Assumptions : Observations in are temporally ordered

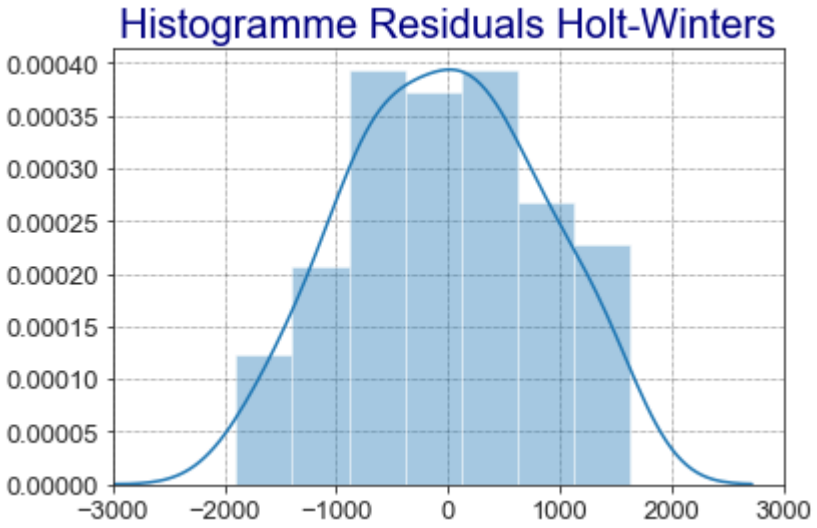
H0: a unit root is present (series is non-stationary).
H1: a unit root is not present (series is stationary).
stat=-1.807, p=0.377
Probably not Stationary

Kwiatkowski-Phillips-Schmidt-Shin
Tests whether a time series is trend stationary or not.

Assumptions : Observations in are temporally ordered

H0: the time series is not trend-stationary.
H1: the time series is trend-stationary.
stat=0.351, p=0.098
Probably not Stationary

Forecast : méthode Holt-Winters



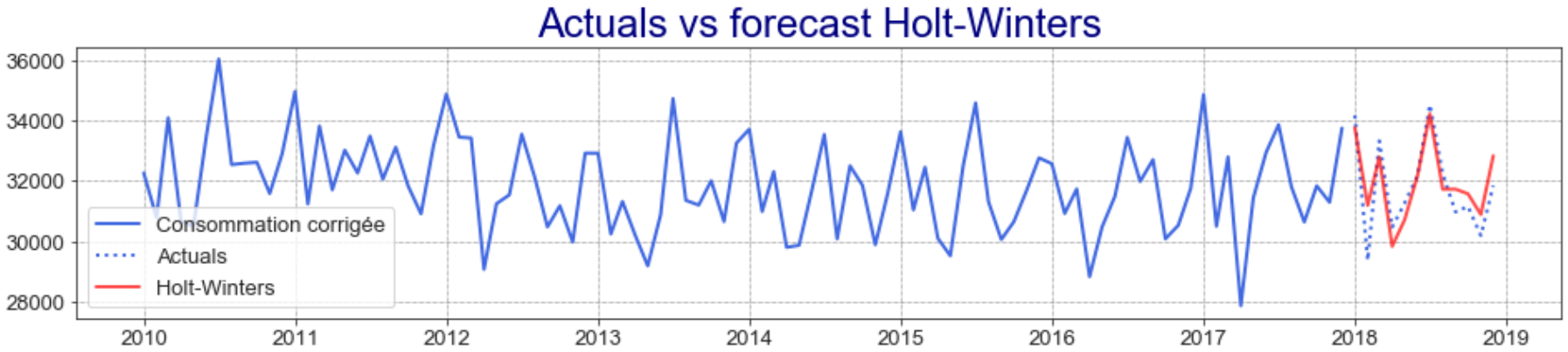
ExponentialSmoothing Model Results

Dep. Variable:	endog	No. Observations:	96
Model:	ExponentialSmoothing	SSE	69190874.005
Optimized:	True	AIC	1326.851
Trend:	Additive	BIC	1367.881
Seasonal:	Additive	AICC	1335.734
Seasonal Periods:	12	Date:	Wed, 22 Jan 2020
Box-Cox:	False	Time:	12:06:22
Box-Cox Coeff.:	None		

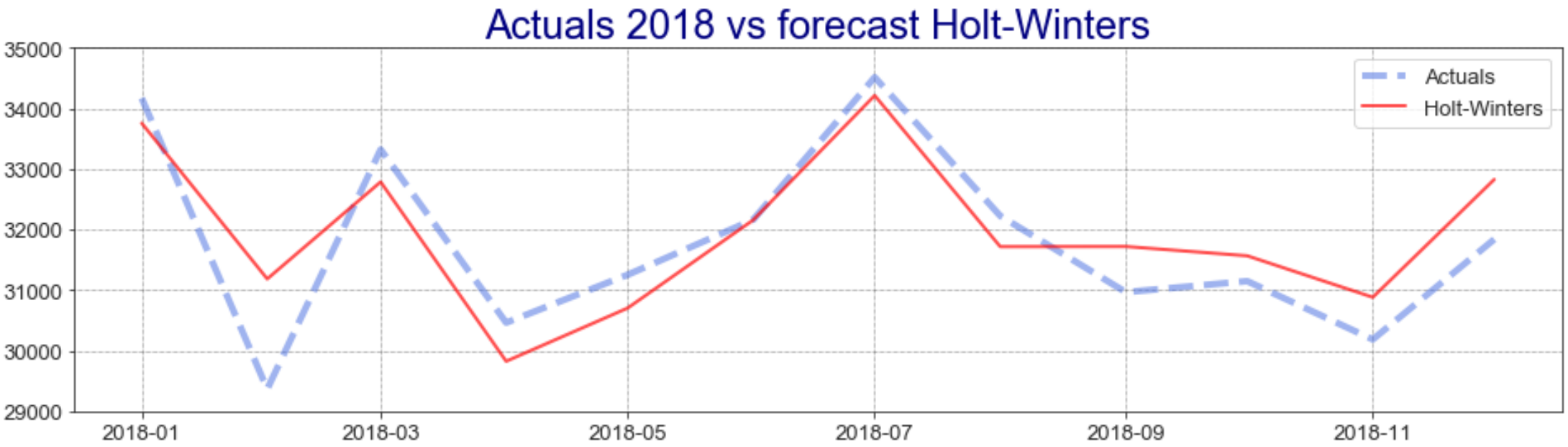
	coeff	code	optimized
smoothing_level	0.1341256	alpha	True
smoothing_slope	3.3302e-20	beta	True
smoothing_seasonal	1.9981e-19	gamma	True

Shapiro-Wilk Normality:

stat=0.984, p=0.312
Probably Gaussian



Performance



Root mean square error (RMSE)

RMSE HOLT-WINTERS: 765.292

Mean absolute percentage error (MAPE)

MAPE HOLT-WINTERS: 2.043

AIC **1326.860**

BIC **1367.890**

Notation :
(p,d,q)(P,D,Q)m

Forecast : Méthode SARIMA

Configurer SARIMA requiert de sélectionner des hyperparamètres pour Trend et Saisonalité

Eléments Trend

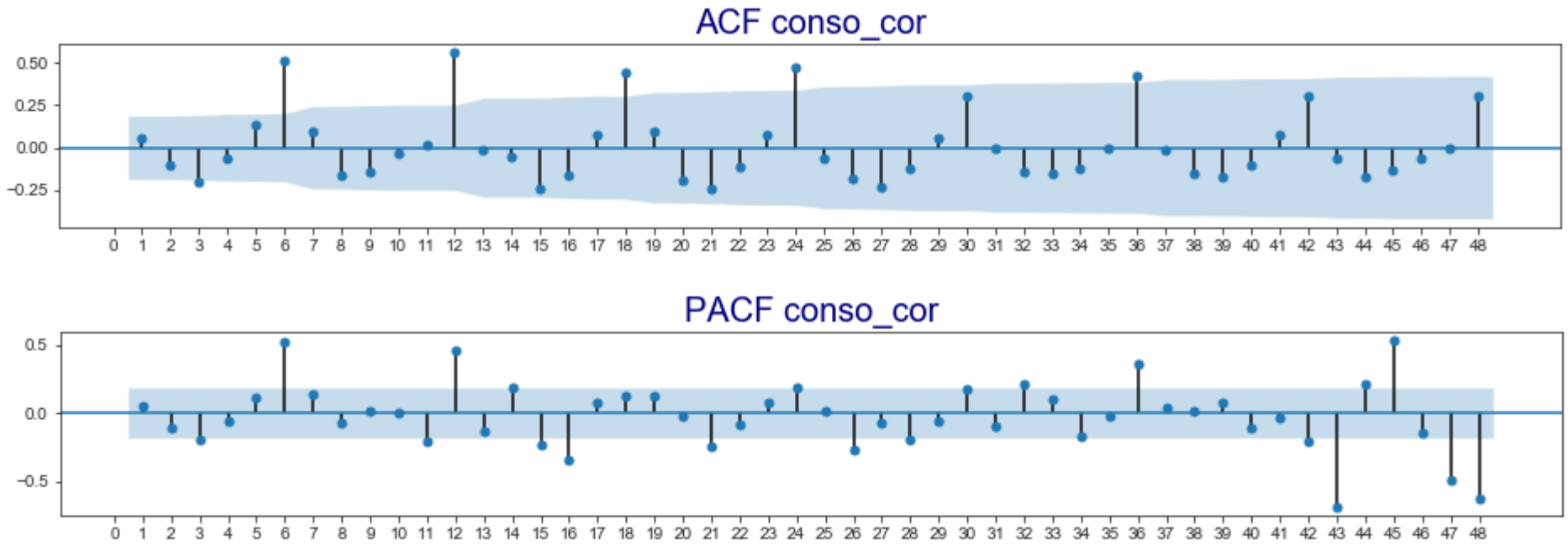
Ce sont les mêmes éléments que pour ARIMA

- p: Trend autoregression order (AR)
- d: Trend difference order.
- q: Trend moving average order (MA)

Eléments saisonnalité

- P: Seasonal autoregressive order.
- D: Seasonal difference order.
- Q: Seasonal moving average order.
- m: nombre d'étapes pour un seule période 'saison'.

ACF et PACF


ACF conso_cor
0.50
0.25
0.00
−0.25
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
PACF conso_cor
0.5
0.0
−0.5
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48

Grille de recherche SARIMA empirique

Identification de l'ordre de différenciation

- d=0 si la série n'a pas de trend visible ou si l'ACF est bas pour tous les lags
- d≥1 si la série a une trend visible ou un ACF positif pour un nombre élevé de lags

Note: Si d=1 semble être la meilleure option, la série a une trend constante. Un modèle avec d=2 assume que la série a une trend variant au cours du temps

Identification des termes AR et MA

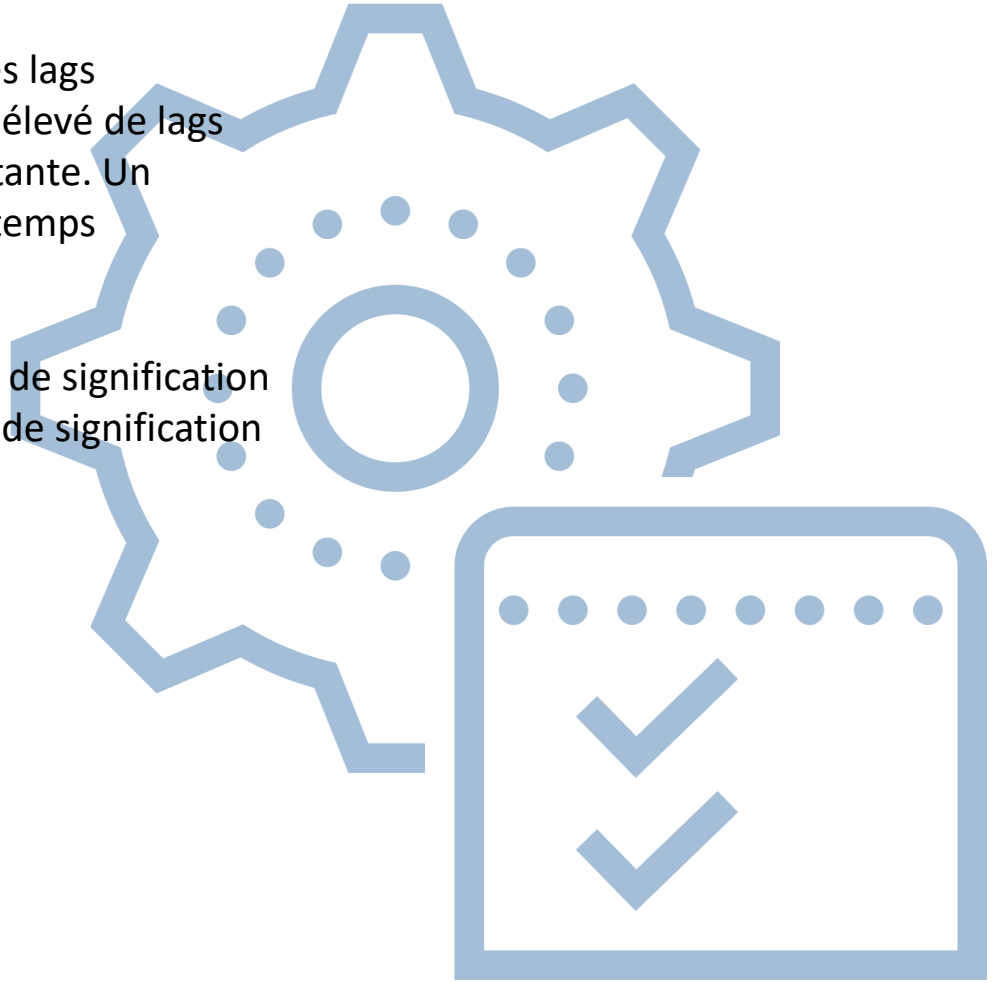
- p est égal au premier lag PACF dont la valeur est supérieure au seuil de signification
- q est égal au premier lag don't la valeur ACF est supérieure au seuil de signification

Identification de la part saisonnière du modèle

- M est égal au lag ACF avec la plus grande valeur
- D=1 si la série a un modèle saisonnier stable au cours du temps
- D=0 si la série a un modèle saisonnier instable au cours du temps

Règles d'or

- d+D≤2
- P≥1 si l'ACF est positif au lag m, sinon P=0.
- Q≥1 si l'ACF est negative au lag m, sinon Q=0



Sélection SARIMA empirique

Eléments Trend du modèle (p,d,q)

- p = 3 (premier lag PACF supérieur au seuil de signification)
- q = 3 (premier lag ACF supérieur au seuil de signification)
- d = 0 (pas de trend = trend hératique)

Eléments saisonniers du modèle (P, D, Q)m

- m = 12 (lag ACF avec la plus haute valeur)
- P ≥ 1 = 2 (ACF est positif lag m, lag 2m significatif)
- Q = 0 (ACF est positif au lag m)
- D ≥ 1 = 2 (on assume que la série a une trend variant au cours du temps)

```
# Initialization
data = df.values
order = (3, 0, 3)
seasonal_order = (2,2,0,12)
```

Résultats méthode empirique

Nous obtenons un modèle correct avec des p-values significatives sauf pour ma.L3

```
                                Statespace Model Results
==============================================================================================
Dep. Variable:                                  y   No. Observations:                   96
Model:             SARIMAX(3, 0, 3)x(2, 2, 0, 12)   Log Likelihood                -641.412
Date:                            Mon, 27 Jan 2020   AIC                           1300.824
Time:                                    06:54:32   BIC                           1321.314
Sample:                                         0   HQIC                          1308.981
                                              - 96
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          1.4720      0.087     16.888      0.000       1.301       1.643
ar.L2         -1.3726      0.131    -10.470      0.000      -1.629      -1.116
ar.L3          0.8269      0.083     10.009      0.000       0.665       0.989
ma.L1         -1.0998      0.195     -5.645      0.000      -1.482      -0.718
ma.L2          1.1029      0.232      4.759      0.000       0.649       1.557
ma.L3         -0.3740      0.205     -1.827      0.068      -0.775       0.027
ar.S.L12      -0.5207      0.062     -8.420      0.000      -0.642      -0.400
ar.S.L24      -0.1838      0.034     -5.404      0.000      -0.250      -0.117
sigma2      4.032e+06   4.62e-09   8.74e+14      0.000    4.03e+06    4.03e+06
```

```python
# Initialization
data = df.values
order = (3, 0, 3)
seasonal_order = (2,2,0,12)
```

SARIMA modèle simplifié

La trend de la série est non-linéaire, on peut considerer qu'elle n'au que peu d'influence sur le forecast SARIMA qui depend essentiellement de la siasonnalité

```
# Initialization
data = df.values
order = (0,0,0)
seasonal_order = (2,2,0,12)
```

Statespace Model Results

Dep. Variable:	y	No. Observations:	96
Model:	SARIMAX(2, 2, 0, 12)	Log Likelihood	-655.861
Date:	Mon, 27 Jan 2020	AIC	1317.722
Time:	06:54:34	BIC	1324.552
Sample:	0	HQIC	1320.442
	- 96		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.S.L12	-0.0235	0.034	-0.697	0.486	-0.090	0.043
ar.S.L24	-0.0282	0.025	-1.138	0.255	-0.077	0.020
sigma2	4.776e+06	3.97e-10	1.2e+16	0.000	4.78e+06	4.78e+06

Modèle Auto SARIMA

pmdarima est un package Python qui détecte le modèle SARIMA optimal comme la fonctionalité auto.sarima dans R

```
auto_arima(df.values, n_jobs = -1,
m=12, max_p = 3, max_q = 3,
seasonal=True, max_P = 2, Q=0,
d=0, D=2, trace=True, alpha = 0.05,
error_action='ignore',
suppress_warnings=True,
stepwise=True)
```

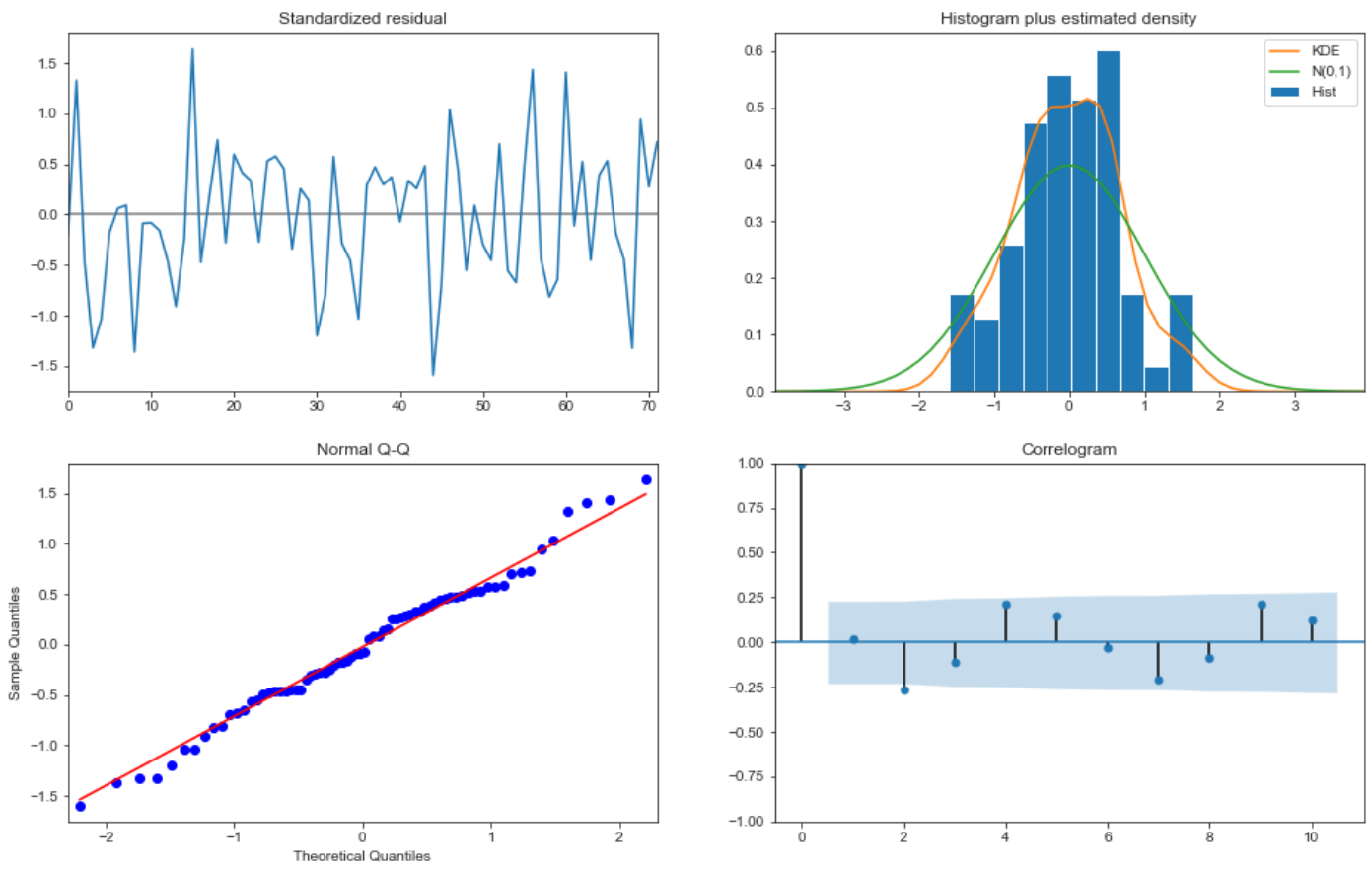
Statespace Model Results

Dep. Variable:	y	No. Observations:	96
Model:	SARIMAX(1, 0, 2)x(0, 2, 1, 12)	Log Likelihood	-635.858
Date:	Mon, 27 Jan 2020	AIC	1283.717
Time:	10:11:11	BIC	1297.377
Sample:	0	HQIC	1289.155
	- 96		
Covariance Type:	opg		

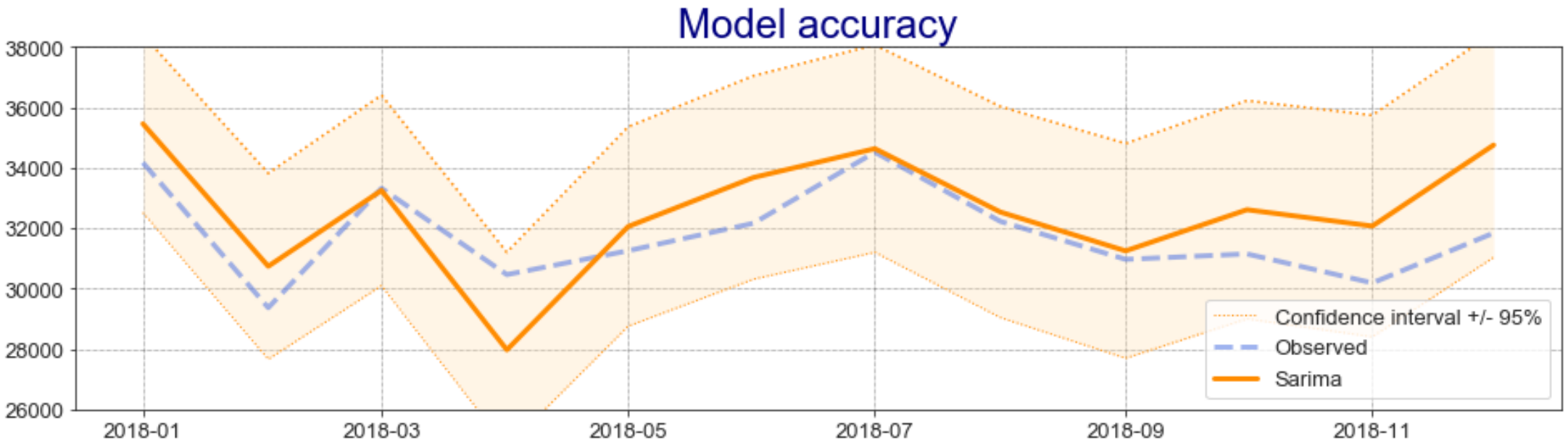
	coef	std err	z	P>|z|	[0.025	0.975]
intercept	-1.6280	39.107	-0.042	0.967	-78.276	75.020
ar.L1	0.9854	0.078	12.641	0.000	0.833	1.138
ma.L1	-0.7966	0.278	-2.862	0.004	-1.342	-0.251
ma.L2	0.0374	0.207	0.181	0.856	-0.367	0.442
ma.S.L12	-0.9275	0.312	-2.977	0.003	-1.538	-0.317
sigma2	4.086e+06	0.000	4.03e+10	0.000	4.09e+06	4.09e+06

Modèle SARIMA retenu

- Les 3 méthodes présentent des résultats relativements similaires.
- Nous retenons Auto SARIMA pour la suite, car il offre le meilleur AIC et dont le degré de complexité n'est pas trop élevé

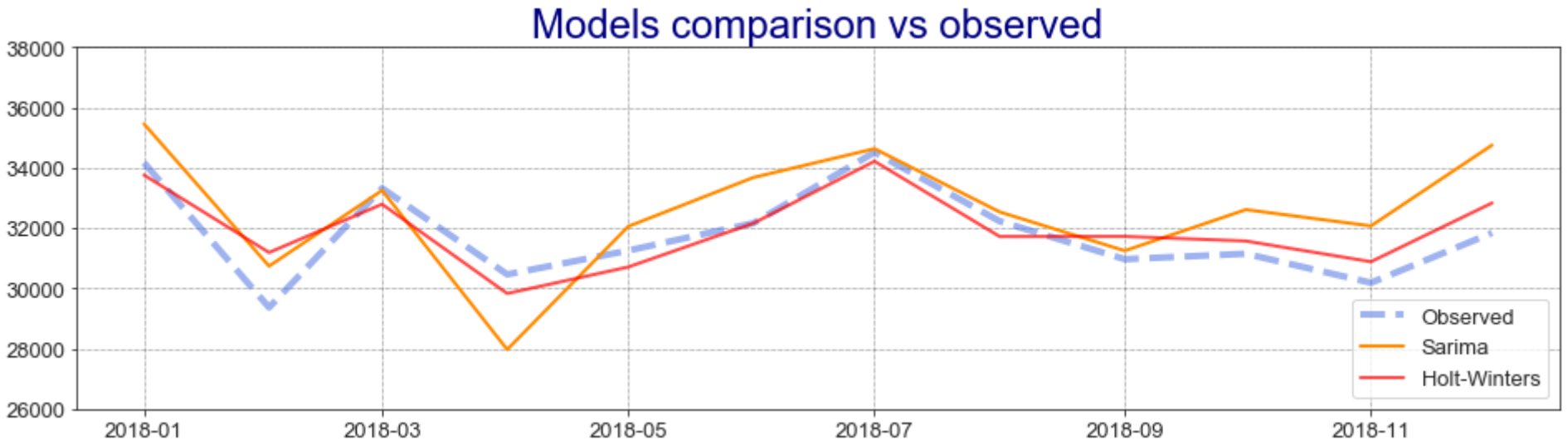


Forecast Auto SARIMA



- Root mean square error (RMSE) : 1501.942
- Mean absolute percentage error (MAPE) : 3.868

Comparaison de la qualité des forecasts




```
RMSE HOLT-WINTERS:  765.904 < RMSE SARIMA: 1501.942
MAPE HOLT-WINTERS:    2.042 < MAPE SARIMA:    3.868
AIC  HOLT-WINTERS: 1326.860 > AIC  SARIMA: 1283.717
BIC  HOLT-WINTERS: 1367.890 > BIC  SARIMA: 1289.377
```

Conclusions

- Nous avons réalisé une étude préliminaire avec les données France et DJU Paris
- Pour obtener des données plus fines localement, maintenant que nous pouvons automatiser la tâche, il faudrait réaliser l'étude pour chaque région et pour chacune des stations de ces régions
- Notre meilleur modèle prédictif est Holt-winters
- Nous nous sommes limités aux données du chauffage pour le DJU, mais nous pourrins inclure la climatisation dans notre modèle

Des questions ?

lgourdon@gmail.com (+34 659 34 05 15)