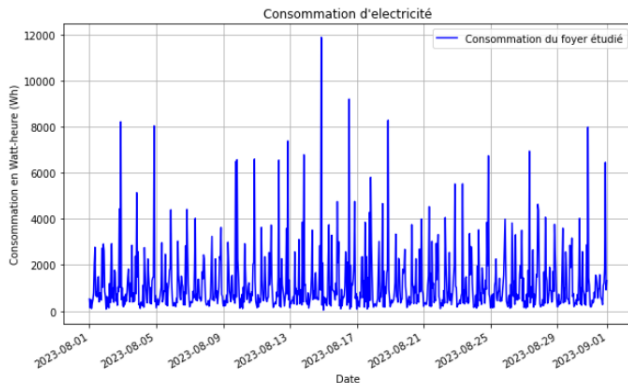


Application des séries chronologiques sur les données Dataset_CB.csv

Youness et Ivanhoé

11 janvier 2024

Introduction



Voici une série chronologique issue de données pseudo-réelles issue d'une consommation EDF d'un foyer.

Problématique : Comment prédire à horizon de 7 jours la consommation électrique du foyer sur la première semaine de septembre 2023 ?

- 1 Partie preprocessing
- 2 Stationnarité et vérification
- 3 Étude des corrélations

Renommer les colonnes et passer au format date

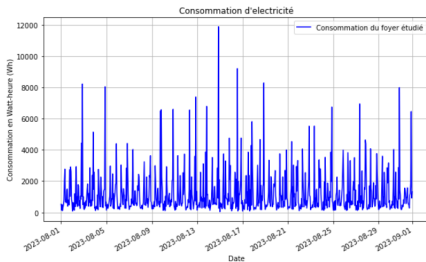
```
EDF = pd.read_csv(path)
EDF.columns = ["DATE", "CONSO"] # renommer col
EDF.DATE = pd.to_datetime(EDF.DATE, format='%d/%m/%y %H:%M')
print("shape = ", EDF.shape)
EDF.head(4)
```

```
shape = (744, 2)
```

Chargement des données

	DATE	CONSO
0	2023-08-01 00:00:00	522.646044
1	2023-08-01 01:00:00	142.889213
2	2023-08-01 02:00:00	497.052422
3	2023-08-01 03:00:00	110.082063

Dataset EDF



Consommation du foyer

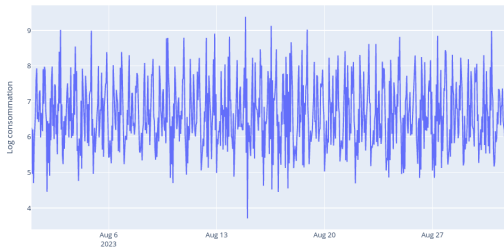
Passage au log pour réduire la variabilité de la série

```
EDF['LOGCONSO'] = np.log(EDF.CONSO)  
EDF.head()
```

	DATE	CONSO	LOGCONSO
0	2023-08-01 00:00:00	522.646044	6.258904
1	2023-08-01 01:00:00	142.889213	4.962070
2	2023-08-01 02:00:00	497.052422	6.208695
3	2023-08-01 03:00:00	110.082063	4.701226
4	2023-08-01 04:00:00	333.036926	5.808253

Dataset EDF

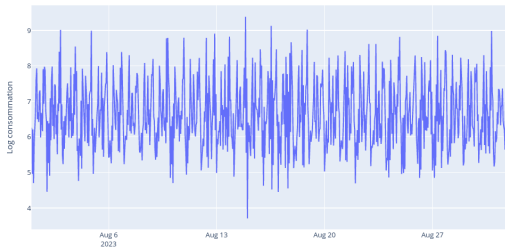
Log-consommation d'énergie du foyer



Log-consommation du foyer

Étude de la stationnarité de la log-consommation

Log-consommation d'énergie du foyer



Log-consommation du foyer

Sans différencier

```
# Testons la non-stationnarité
TestA = adfuller(EDF.LOGCONSO) # Test ADF rejeté
print("ADF p-val : ", TestA[1])

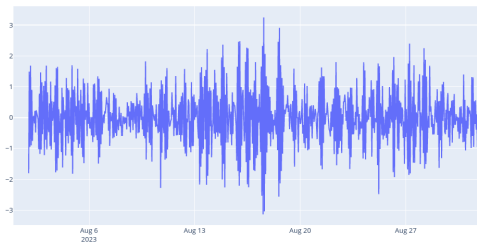
# Testons la stationnarité
TestK = kpss(EDF.LOGCONSO) # Test KPSS non rejeté
print("KPSS p-val : ", TestK[1])# Test non rejeté
```

ADF p-val : 0.0
KPSS p-val : 0.1

Tests ADF et KPSS

Étude de la stationnarité de la log-consommation avec une différenciation saisonnière

Différenciation saisonnière de la Log-consommation d'énergie du foyer



Différentielle saisonnière
log-consommation du foyer

```
# On va différencier une fois pour regarder les incréments à 24 heures près
EDF["diff24"] = EDF.LOGCONSO.diff(24)
fig = px.line(EDF, x=DATE, y="diff24", labels={'DATE': 'Date', 'diff24': ''})
fig.update_layout(title="Différenciation saisonnière de la Log-consommation d'énergie du foyer")
fig.show()
```

Différenciation D=24

Avec différenciation saisonnière s=24

```
# Testons la non-stationnarité
TestA = adfuller(EDF.diff24.dropna()) # Test ADF rejeté au seuil de 5%
print("ADF p-val : ", TestA[1])

# Testons la stationnarité
TestK = kpss(EDF.diff24.dropna()) # Test KPSS non rejeté
print("KPSS p-val : ", TestK[1])# Test non rejeté

ADF p-val : 0.00022114078230846984
KPSS p-val : 0.1
```

Tests ADF et KPSS

Étude des corrélations sur la série temporelle

```
ACF = plot_acf(EDF.LOGCONSO, lags=70, alpha=0.05, title="ACF avec lags=70")  
PACF = plot_pacf(EDF.LOGCONSO, lags=70, alpha=0.05, title="PACF avec lags=70")  
  
ACF = plot_acf(EDF.LOGCONSO, lags=26, alpha=0.05, title="ACF avec lags=26")  
PACF = plot_pacf(EDF.LOGCONSO, lags=26, alpha=0.05, title="ACF avec lags=26")
```

