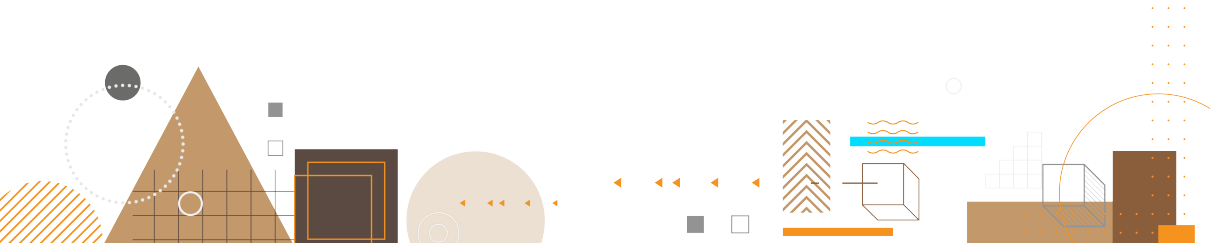


# Présentation du Projet

Développement logiciel

Thibault FERRETTI, Pauline DUSFOUR-CASTAN , Sarah MATOUB, Mathieu LE-SEACH



## 1 Introduction

## 2 Prédiction

- Prédiction consommation d'électricité
- Prédiction autres sources

## 3 Visualisation

- Création des différentes cartes
- Distributions des consommations
- Partie interactive
- Exemple d'utilisation

# Introduction

## Objectifs :

- ⇒ prédiction d'électricité
- ⇒ visualisation de la consommation



- Qu'est ce qu'une série temporelle?
- Qu'est ce que le modèle additif?
- Qu'est ce que le modèle Prophet?
- Comment fait-il pour faire ces prévisions?

Série temporelle : suite de données indexées par le temps.

Modèle additif :

$$Y(t) = g(t) + s(t) + \epsilon(t)$$

avec respectivement :

$g(t)$  : la tendance .

$s(t)$  : une ou plusieurs composantes saisonnières.

$\epsilon(t)$  : l'erreur.

Prophet : modèle de prévisions des séries temporelles.

$$Y(t) = g(t) + s(t) + h(t) + \epsilon(t)$$

avec respectivement :

$g(t)$  : la tendance .

$s(t)$  : une ou plusieurs composantes saisonnières.

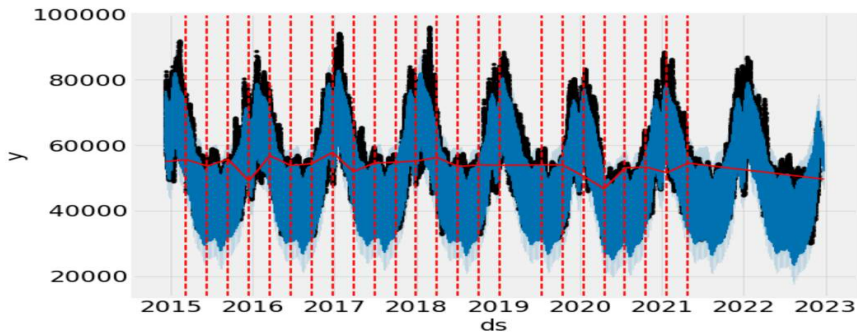
$h(t)$  : effet des vacances.

$\epsilon(t)$  : l'erreur.

# Prédiction consommation d'électricité

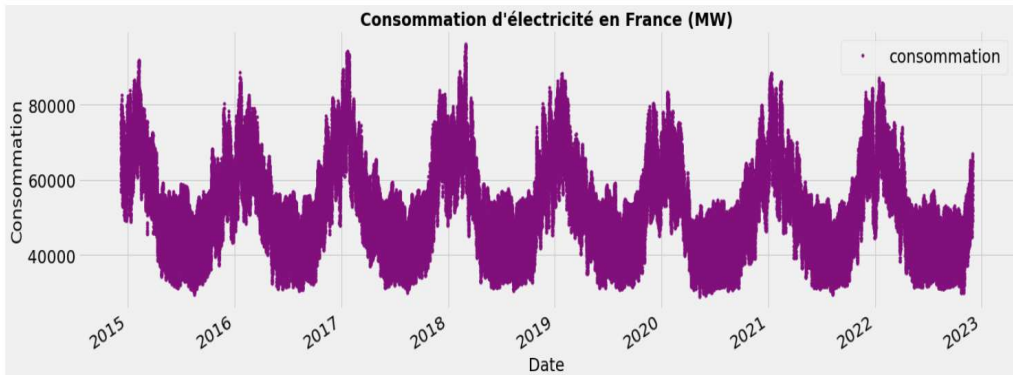
Comment fait-il pour faire ces prévisions?

```
from prophet.plot import add_changepoints_to_plot  
fig = m.plot(forecast)  
a = add_changepoints_to_plot(fig.gca(),m, forecast)
```



# Prédiction de la consommation d'électricité

Données de la consommation d'électricité en France :

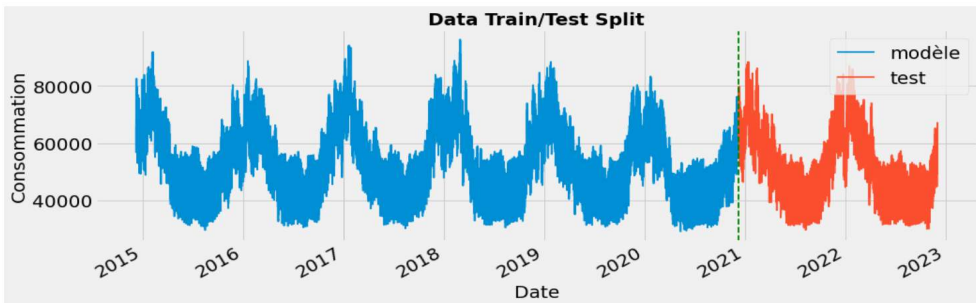




# Prédiction consommation d'électricité

Création du modèle/test :

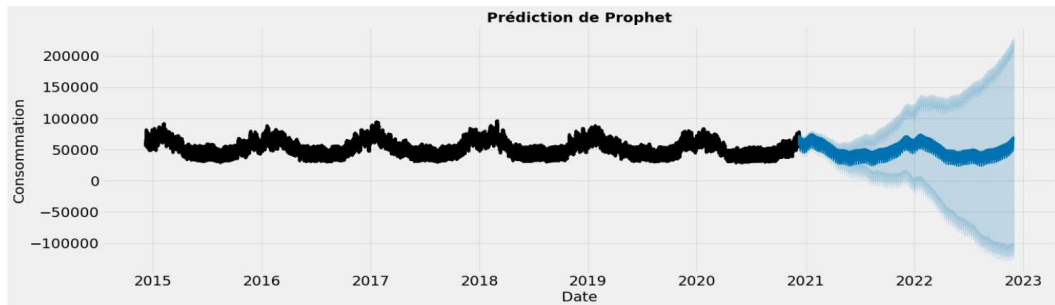
```
train = df.iloc[df.index < '2021-12-08']  
train_pr = train.reset_index()  
test = df.iloc[df.index >= '2021-12-08']  
test_pr = test.reset_index()
```



# Prédiction consommation d'électricité

Application du modèle :

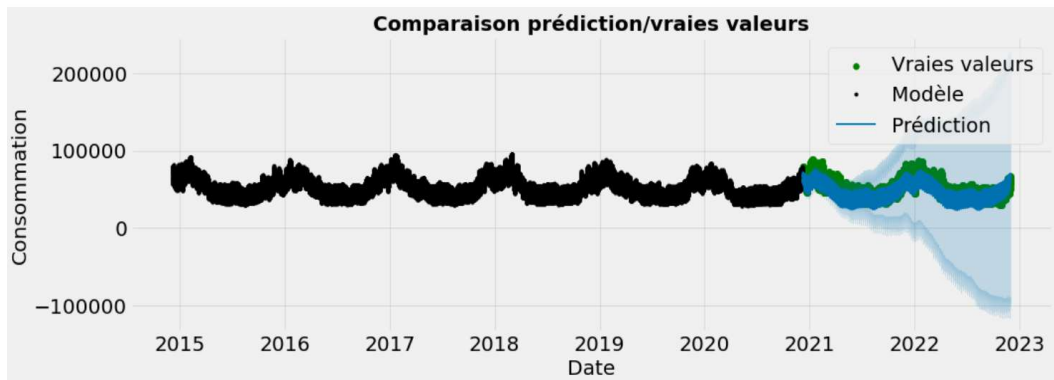
```
from prophet import Prophet  
model = Prophet()  
model.fit(train_pr)  
future = model.make_future_dataframe(test_pr)  
test_frc = model.predict(future)
```



# Prédiction consommation d'électricité

Prédiction vs vraies valeurs:

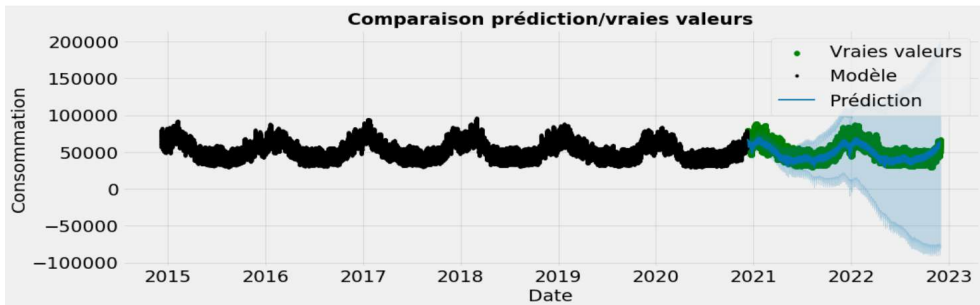
```
f, ax = plt.subplots(figsize=(15, 5))  
ax.scatter(test.index, test, color='g')  
fig = model.plot(test_frc, ax=ax)
```



# Prédiction consommation d'électricité

Prise en compte de l'effet des "vacances":

```
model_vac = Prophet(daily_seasonality = False)
model_vacc.add_country_holidays(country_name = "FR")
model_vac.fit(train_pr)
```



Prédiction sur 10 jours à partir du 29 novembre :

```
from prophet import Prophet
    m = Prophet()
    m.fit(df_cons)
    f = m.make_future_dataframe(periods=48*10 , freq='30min',
    include_history=False)
    predic = m.predict(f)
```

Prédiction finale :

Date et heure	Prévision	Intervalle de confiance (à 95%)	Vraie valeur
2022-12-08 00:00:00	62770.97	[57346.92 ,68357.66]	64 589
2022-12-08 00:30:00	61537.83	[56225.18, 66887.56]	64 008
2022-12-08 01:00:00	60203.41	[54602.52, 65590.53]	61 178
2022-12-08 01:30:00	58931.79	[53544.44, 64333.79]	61 427
2022-12-08 02:00:00	57912.61	[52765.69, 63608.31]	61 415
2022-12-08 02:30:00	57321.09	[52002.38, 62917.20]	61 106

Données de la consommation d'électricité le 8 décembre 2022

But : prédire la consommation de différentes sources d'énergie



Gaz, Fioul, Charbon, Nucléaire, Eolien, Hydraulique, Solaire

# Prédiction des sources d'électricité

Première étape : création des bases de données + nettoyer et trier les données

Date et Heure		Gaz (MW)	Fioul (MW)	Eolien (MW)	Hydraulique (MW)
Date					
2021-12-08	2021-12-08T04:30:00+01:00	4861.0	112.0	9446.0	5024.0
2021-12-08	2021-12-08T06:30:00+01:00	6696.0	116.0	8426.0	6346.0
2021-12-08	2021-12-08T11:00:00+01:00	7629.0	115.0	6281.0	9006.0
2021-12-08	2021-12-08T18:30:00+01:00	8427.0	134.0	5318.0	10684.0
2021-12-08	2021-12-08T20:00:00+01:00	8254.0	134.0	5761.0	9751.0
...	...	...	...	...	...
2012-12-08	2012-12-08T14:00:00+01:00	3456.0	489.0	1077.0	7346.0
2012-12-08	2012-12-08T18:30:00+01:00	3504.0	488.0	854.0	10285.0
2012-12-08	2012-12-08T19:30:00+01:00	3503.0	487.0	851.0	9839.0
2012-12-08	2012-12-08T21:00:00+01:00	3532.0	489.0	878.0	7414.0
2012-12-08	2012-12-08T21:30:00+01:00	3582.0	489.0	906.0	7205.0



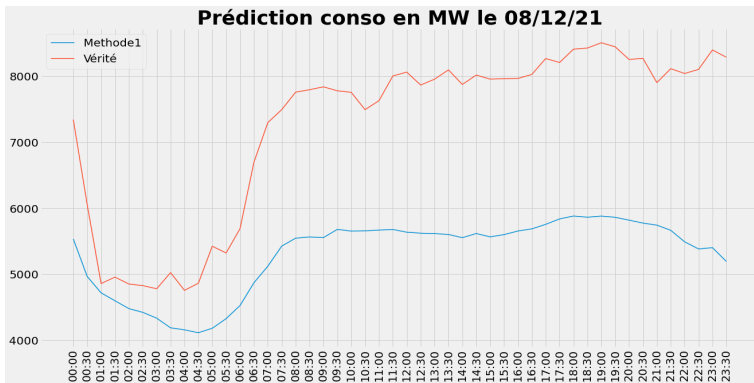
# Prédiction des sources d'électricité

## Seconde étape : Application de la méthode 1

⇒ Méthode de la moyenne empirique

⇒ Utilisation de

```
df.groupby(["Heure"])["Gaz(MW)"].mean()
```

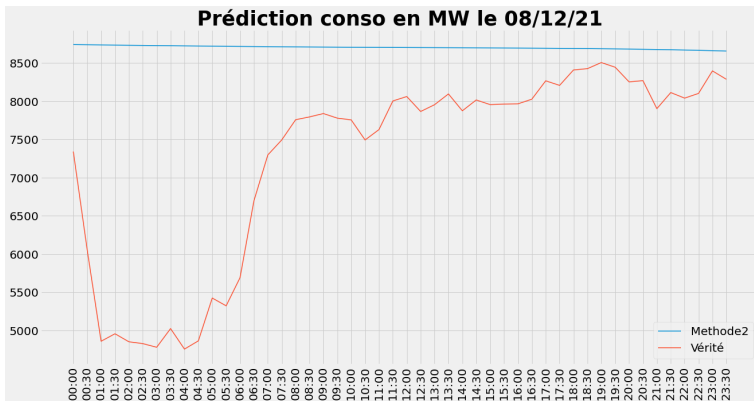


# Prédiction des sources d'électricité

## Troisième étape : Application de la méthode 2

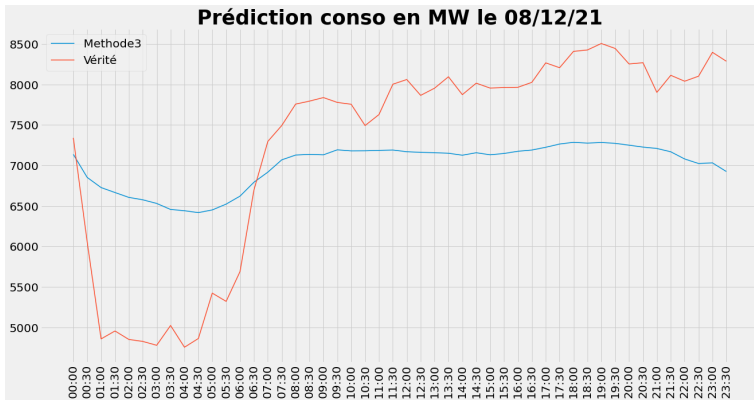
⇒ Méthode Prophet

⇒ Utilisation des séries temporelles



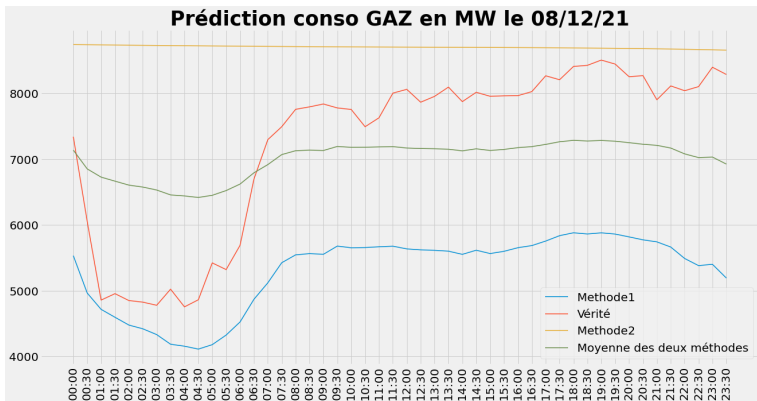
Quatrième étape : Application de la méthode 3

⇒ Moyenne des deux méthodes



# Prédiction des sources d'électricité

⇒ Comparaison des différentes méthodes



En moyenne :

$$|\text{vraie valeur} - \text{methode}| = \begin{cases} 1940 \text{ MW} & \text{methode 1} \\ 1461 \text{ MW} & \text{methode 2} \\ 971 \text{ MW} & \text{methode 3} \end{cases}$$

⇒ La méthode qui se rapproche le plus de la vérité est la méthode 3

# Prédiction des sources d'électricité

Ci dessous, les prédictions pour le 8 décembre 2022:

	Heure	moyenne	methode1	methode2
0	00:00	6084.163005	5712.7	6455.626011
1	00:30	5763.990576	5072.6	6455.381151
2	01:00	5592.295730	4729.3	6455.291460
3	01:30	5543.034475	4630.7	6455.368950
4	02:00	5484.562425	4513.5	6455.624851
5	02:30	5458.034771	4460.0	6456.069541
6	03:00	5416.356248	4376.0	6456.712495
7	03:30	5362.981110	4268.4	6457.562219
8	04:00	5336.663099	4214.7	6458.626198
9	04:30	5309.755199	4185.6	6459.919915

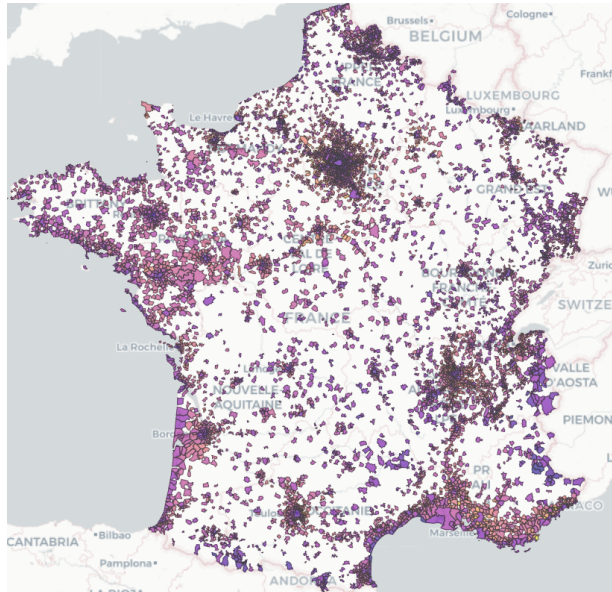
Principaux Packages utilisés:

- ① Pandas: Traitement des données
- ② Plotly: Aspect graphique
- ③ Dash: Aspect interactif

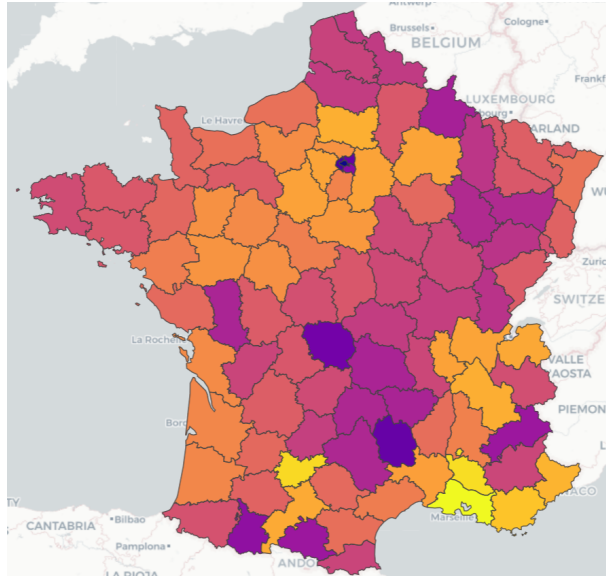
- ① Traitement des données
- ② Liaison entre le fichier geojson et le dataframe
- ③ Affichage de la carte



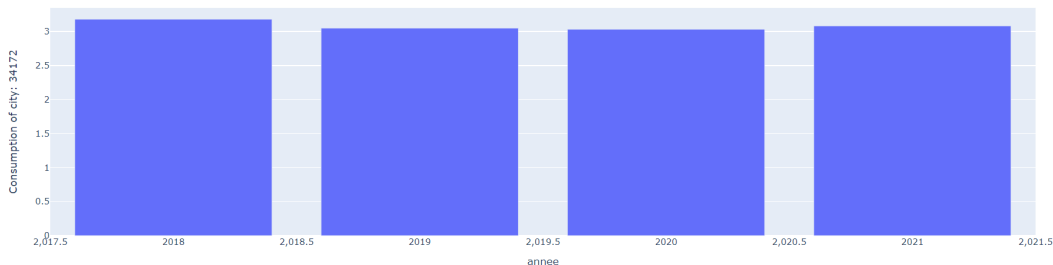
# Création des différentes cartes



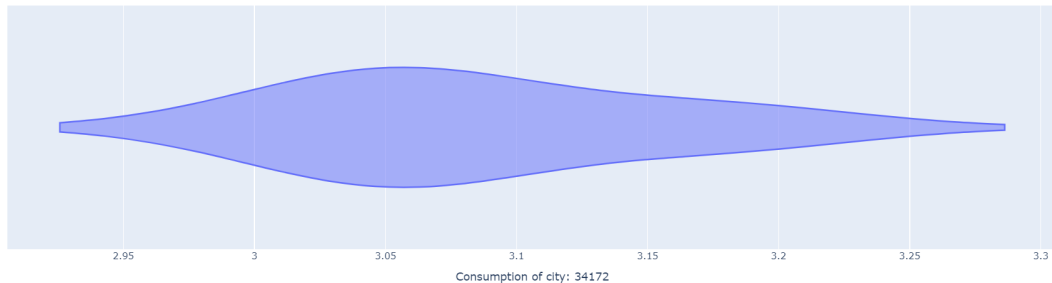
# Création des différentes cartes



Création d'une classe "Ville" permettant de retourner différents types de distribution des consommations au cours des quatre années données:

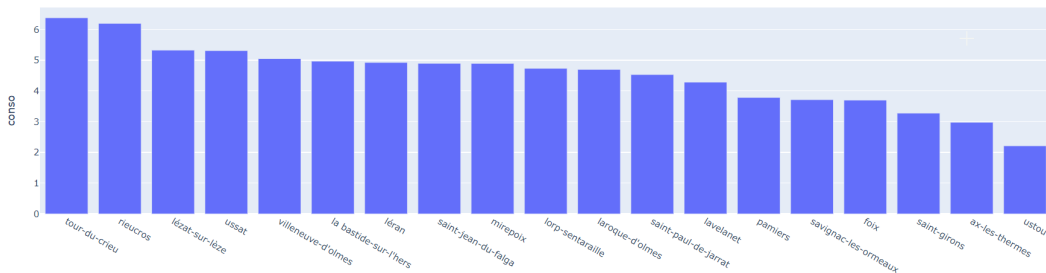


Violon plot des consommations sur les quatre années:



# Distributions des consommations

Histogramme des villes du département rangées pas ordre décroissant de consommation. Exemple de l'Ariège:



Menu déroulant des départements et affichage des villes avec la consommation la plus élevée/faible.

Exemple de l'Ariège:

Occitanie

×

▼

Ariège

×

▼

The city with the maximum consumption is: ('tour-du-crieu', 6.378) The city with the minimum consumption is: ('ustou', 2.211)

Association du click sur la carte des villes à la distribution au cours des quatre années données:

```
def update_plot(option_slctd, clickdata):  
  
    code = clickdata["points"][0]["location"]  
  
    if option_slctd == "violin":  
        fig = City(code).violin()  
  
    return fig
```

L'application a été hébergé sur le site web `pythonanywhere.com`:

► <http://thibaultf02.pythonanywhere.com>



Merci pour votre écoute.

