

Rapport de projet de programmation informatique

Membres du groupe : Fridhi Iliès, Léger Maureen
Sujet 2

I. Introduction

a) Présentation et analyse des données CSV

Les données sont des mesures de différents capteurs d'identifiants allant de 1 à 6. Nous disposons, sous format CSV, de la mesure de la température ambiante (°C), de l'humidité relative (%), du niveau sonore (dBA), du niveau lumineux (lux), de la quantité de CO₂ (ppm). Nous avons également les instants auxquelles les mesures ont été effectuées.

b) Présentation du sujet

L'objectif de la programmation est de mesurer les similarités des capteurs pour chaque dimension. Pour cela, il faudra un algorithme permettant de les mesurer automatiquement et de les afficher sur une courbe.

II. Méthodologie employée / Algorithme

- Appel aux bibliothèques nécessaires
- Traitement des données CSV : importer et extraire
 - **chemin** : adresse du CSV
 - **données** : variable : création d'une liste avec les lignes du CSV
- Mise en place de moyens
 - **extract(data,cat,sen)** :
fonction qui renvoie la liste de chaînes de caractères correspondant au numéro de capteur et au paramètre désiré;
arguments d'entrée : data – liste - liste des données , cat – str – nom du paramètre , sen – int – identifiant du capteur ;
descriptif ; on vérifie que la catégorie voulue existe dans data ; on distingue les cas pour les paramètres mesurés et le temps ; on utilise pd.Timestamp() pour convertir les dates dans un modèle compréhensible par matplotlib.pyplot. Si on rentre None, on considère tous les capteurs.
 - **interv(plage,tinf,tsup)** :
fonction qui renvoie la liste d'indice correspondant à la position de tinf et tsup ;
arguments d'entrée : plage - list- liste des dates (jour et heures) , tinf – str – instant inférieur de la plage d'échantillonnage , tsup – str – instant supérieur de la plage d'échantillonnage ;

- **tracer(data, cat,sen,tinf,tsup) :**
procédure qui fait afficher sur un graphique, à un capteur donné, le paramètre choisi en fonction du temps, sur l'intervalle [tinf,tsup] ;
arguments d'entrée :
data – liste - liste des données , cat – str – nom du paramètre , sen – int – identifiant du capteur, tinf – str – instant inférieur de la plage d'échantillonnage , tsup – str – instant supérieur de la plage d'échantillonnage ;

- Calcul de l'indice de l'humidex

- **formule_humidex(temp,hum) :**
fonction qui renvoie un flottant qui est l'indice de l'humidex associé à une température donnée temp et une humidité relative donnée hum ;
Les arguments d'entrée sont des floatants (temp et hum).On vérifie les hypothèses pour calculer cet indice.
TR désigne le point de rosée, d'après la formule de Heinrich Gustav **Magnus-Tetens**, on a :
 $a,b= 17.27,237.7$
 $\text{alpha}= (\text{a}*\text{temp})/(\text{b}+\text{temp})+\ln(\text{hum})$
 $\text{TR}=(\text{b}*\text{alpha})/(\text{a}-\text{alpha})$
où :
 $0 < \text{temp} < 60 \text{ }^{\circ}\text{C}$; ce qui est vrai pour tous les capteurs et toutes les températures
 $1\% < \text{hum} < 100\%$
 $0 < \text{TR} < 50 \text{ }^{\circ}\text{C}$
- **indice_humidex(ltemp,lhum,ltime,tinf,tsup):**
fonction qui renvoie la liste des indices de l'humidex sur une durée au cours du temps et qui affiche ses indices de l'humidex en fonction du temps ;
argument d'entrée : ltemp – list- liste des valeurs de température, lhum – list – liste des valeurs de l'humidité relative, ltime – list – liste des dates

- Rédaction des fonctions statistiques

- Fonctions statistiques basiques
- Fonctions statistiques appliquées
- **evol_distrib_freq(l):** on cherche à obtenir une distribution de fréquences pour chaque valeur de la liste. La liste de départ n'est pas forcément triée, on cherche à savoir quelle fréquence prend chaque valeur dans la liste. On obtient une liste ordonnée croissante pour les x avec les fréquences associées.

action_stat(param,data,cat,sen,*btemp) : à partir d'un paramètre souhaité, d'une base de données, d'un capteur défini (sinon tous), l'argument temporel est optionnel (*), on calcule les différents paramètres statistiques.

action_stat2(param,data,cat1,cat2,sen,*btemp) : on généralise action_stat(param,data,cat,sen,*btemp) et on peut calculer les paramètres statistiques pour un paramètre donné pour deux capteurs différents.

- Calcul de l'indice de corrélation de Pearson entre un couple de

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}$$

variables :

III. Programmation

a) Utilisation de GitHub

GitHub est un outil collaboratif qui permet de contribuer à un projet. La démarche est la suivante :

- On copie un projet sur notre espace GitHub
- On crée une branche thématique à partir d'un master
- On effectue nos modifications et nos améliorations
- Le propriétaire peut choisir de refuser les modifications externes ou de les fusionner avec le projet d'origine
- On met à jour notre version du projet en récupérant les dernières modifications à partir du projet d'origine.

GitHub est également un outil de type source documentaire. Nous avons pu nous inspirer : <https://github.com/wannesm/dtaidistance/tree/982d1dbf4278ac421a693744950a0499d11a9dde>.

b) Programme final

Voici le lien GitHub : https://github.com/lgmaureen23/python_rendu

Nom du programme final : « »

Il y a la nécessité de stocker le programme Python dans le même dossier que le fichier CSV pour lancer le programme (lecture d'un même chemin).

IV. Justifications

a) Problématiques rencontrées

La première difficulté se trouve dans la représentation graphique du temps (format, lecture, affichage) dans la console Python via matplotlib.pyplot. Aussi, il a été difficile de faire appel au programme depuis le terminal.

b) Explication des paramètres statistiques

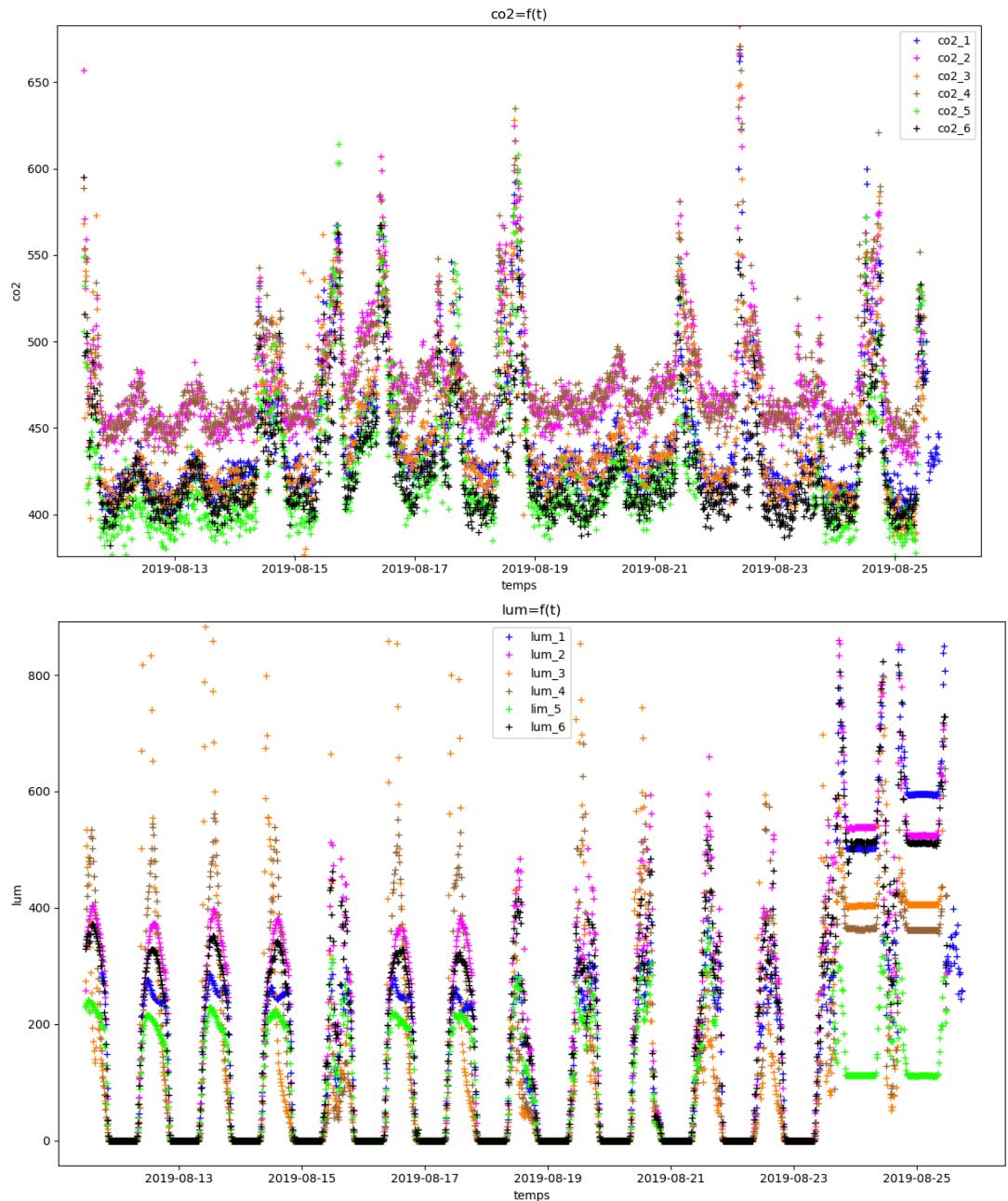
Les paramètres statistiques sont calculés à partir des formules connues. La moyenne arithmétique correspond à une moyenne empirique, la plus simple à analyser.

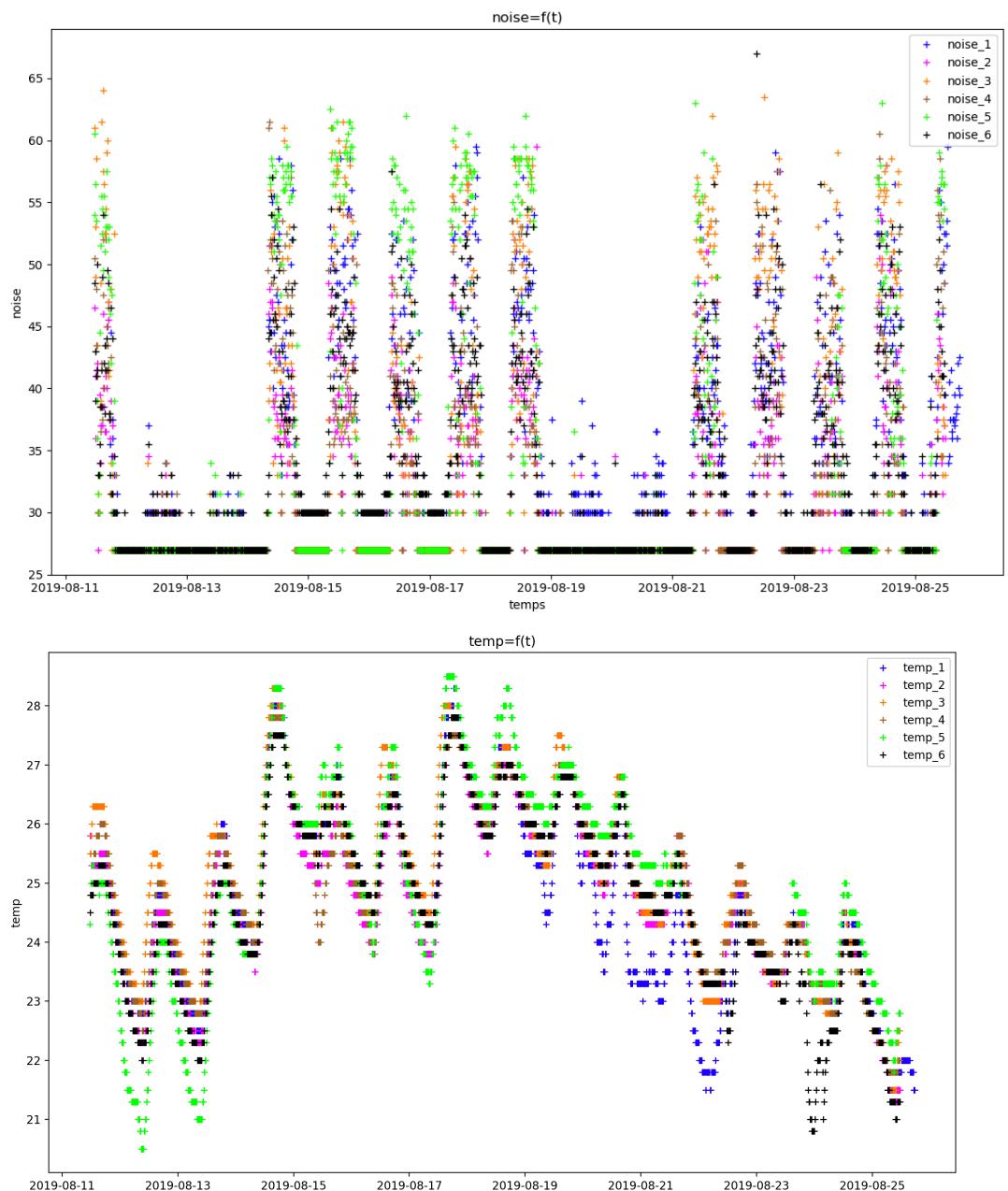
Lorsque l'on veut afficher les moyennes, on utilise la fonction de lissage, on effectue la moyenne sur un intervalle $(k+1)/2+1$, avec k le nombres de termes. Le principe est le suivant : on calcule une moyenne locale à l'aide des voisins proches, selon le principe de la moyenne mobile. Lors du calcul de la moyenne, on atténue donc les variations, ce qui nous oriente donc vers une certaine moyenne.

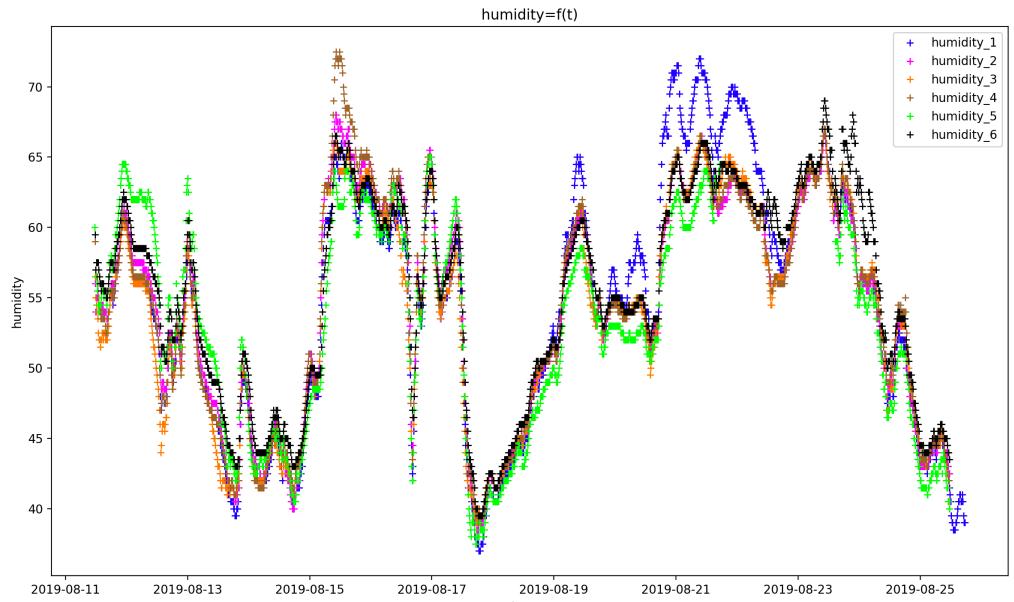
La corrélation est un outil pour mesurer le degré de similarité entre des séries temporelles et mesurer le degré de linéarité entre deux variables.

V. Réponse de la partie commune

a) Affichage des courbes montrant l'évolution des variable en fonction du temps







Analyse qualitative des courbes sur le mois de novembre 2019

Concernant **temp=f(t)** et **humidity=f(t)** avec t désignant le temps, les courbes pour tous les capteurs se superposent quasiment. On peut donc dire que la température et l'humidité sont uniformes dans le bâtiment. Cela peut nous renseigner sur les capacités énergétiques du bâtiment.

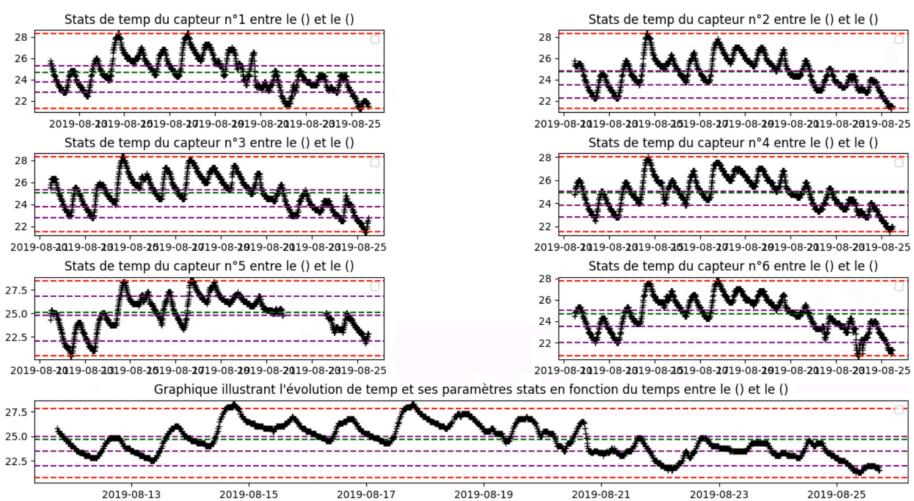
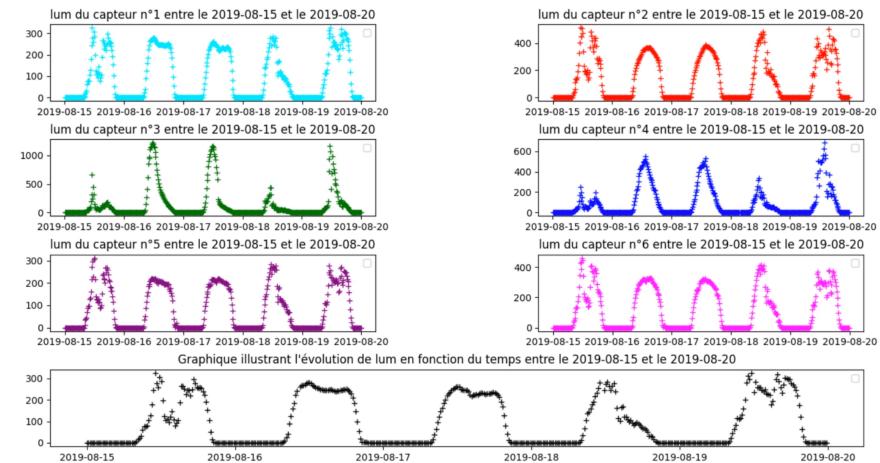
Concernant **noise=f(t)** :

- Du 11 au 14, cela correspond au week-end, on a un niveau sonore ambiant du à l'environnement extérieur.
- Du 15 au 19, on a une alternance de palier de nuit et de niveau sonore plus élevé. Le niveau sonore maximal n'est pas dangereux pour les utilisateurs pour une exposition journalière de 8 heures.
- Du 19 au 21, le bâtiment semble fermé pendant une durée, probablement des vacances ou des congés.
- Du 23 au 25, le bâtiment semble ouvert pendant une période de week-end : un service nettoyage d'où un niveau sonore plus élevé avec l'utilisation des appareils.

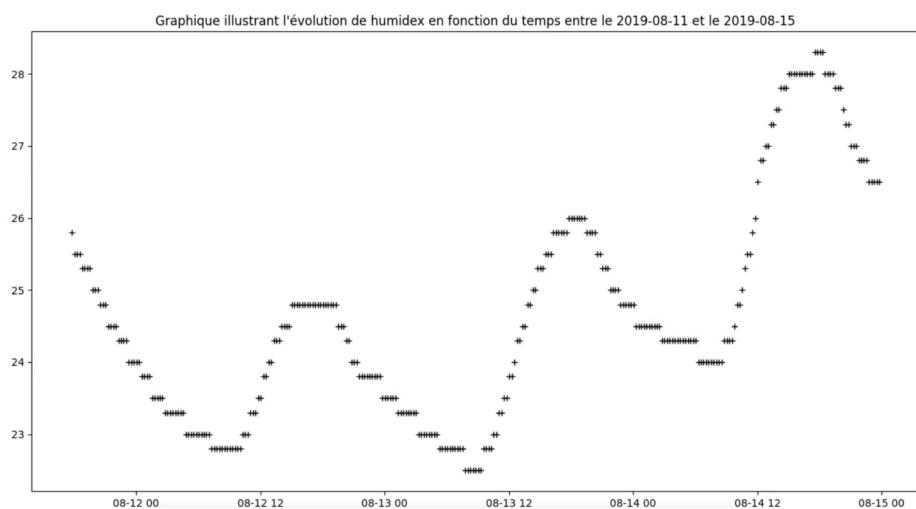
Concernant **co2=f(t)**, on a des « similarités » de comportement entre les capteurs : (1,3),(2,4),(5,6).

Concernant **lum=f(t)**, les paliers bas correspondent à des soirées. Tous les capteurs sont assez similaires en journées avec une intensité variable. Cela doit dépendre de la position et de l'orientation du capteur dans la pièce, si le capteur est situé à proximité d'une source lumineuse (lampe) ou à proximité d'une fenêtre (variable selon la météo), du 13 au 23. Du 23 au 26, on a des paliers bas de nuit beaucoup élevés que les anciens cela doit être due à la lumière artificielle allumée, ce qui peut être corrélé avec le nettoyage.

- b) Exemple d'affichage des valeurs statistiques



c) Calcul de l'indice de l'Humidex



d) Calcul de l'indice de corrélation entre un couple de variables

Le coefficient de corrélation linéaire donne une mesure de l'intensité et du sens de la relation linéaire entre deux variables. Il est compris entre -1 et 1.

Si on est proche de 1, alors, plus la relation linéaire positive entre les variables est forte.

Si on est proche de -1, alors, plus la relation linéaire négative entre les variables est forte.

Si on est proche de 0, alors, plus la relation linéaire entre les variables est faible.

Capteur 1	temp	humidity	co2	lum	noise
temp	0,999	-0,310	0,241	-0,202	0,176
humidity		0,999	0,001	-0,236	-0,066
co2			0,999	0,265	0,721
lum				0,999	0,155
noise					0,999

Capteur 2	temp	humidity	co2	lum	noise
temp	0,999	-0,220	0,116	-0,106	0,120
humidity		0,999	0,191	-0,180	0,012
co2			0,999	0,177	0,605
lum				0,999	0,398
noise					0,999

Capteur 3	temp	humidity	co2	lum	noise
temp	0,999	-0,261	0,307	-0,099	0,202
humidity		0,999	0,095	-0,194	0,029
co2			0,999	0,152	0,733
lum				0,999	0,160
noise					0,999

Capteur 4	temp	humidity	co2	lum	noise
temp	0,999	-0,213	0,098	-0,141	0,065
humidity		0,999	0,088	-0,192	-0,026
co2			0,999	0,150	0,628
lum				0,999	0,336
noise					0,999

Capteur 5	temp	humidity	co2	lum	noise
temp	0,999	-0,253	0,340	0,097	0,263
humidity		0,999	-0,010	-0,106	-0,062
co2			0,999	0,404	0,762
lum				0,999	0,560
noise					0,999

Capteur 6	temp	humidity	co2	lum	noise
temp	0,999	-0,240	0,164	-0,282	0,144
humidity		0,999	0,008	0,081	0,019
co2			0,999	0,265	0,673
lum				0,999	0,428
noise					0,999

VI. Réponse au sujet 2

a) Analyse

La mesure de similarité, selon Jaccard, est une métrique pour comparer la similarité entre deux échantillons. L'indice de Jaccard est le rapport entre le cardinal de l'intersection des ensembles considérés et le cardinal de l'union considéré.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

Dans le domaine de données énergétiques, des études portent sur la comparaison des mesures à différents endroits d'un bâtiment. Pour résoudre le problème de distorsion dans les séries temporelles, on s'intéresse à la méthode qui se base sur la définition de la distance de déformation temporelle « **Dynamic time Warping** » qui considère que le temps est élastique et non linéaire.

Le principe de la distance vise à mettre en correspondance les sous-séquences qui se « ressemblent » même si elles ne correspondent pas à un même intervalle de temps.

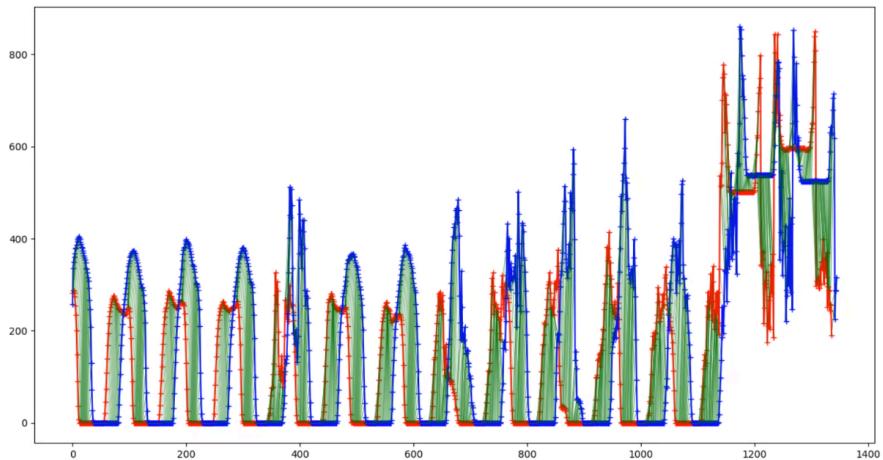
Lorsque l'on compare deux séries temporelles L1 et L2, de longueur respective n et m, on va répliquer les valeurs jusqu'à obtenir la meilleure correspondance. La dimension de la matrice associée est n*m.

Pour relier les points de L1 et de L2, il faut trouver le chemin qui minimise la distance cumulée. Ce principe peut engendré des alignements non désirables.

b) Détail

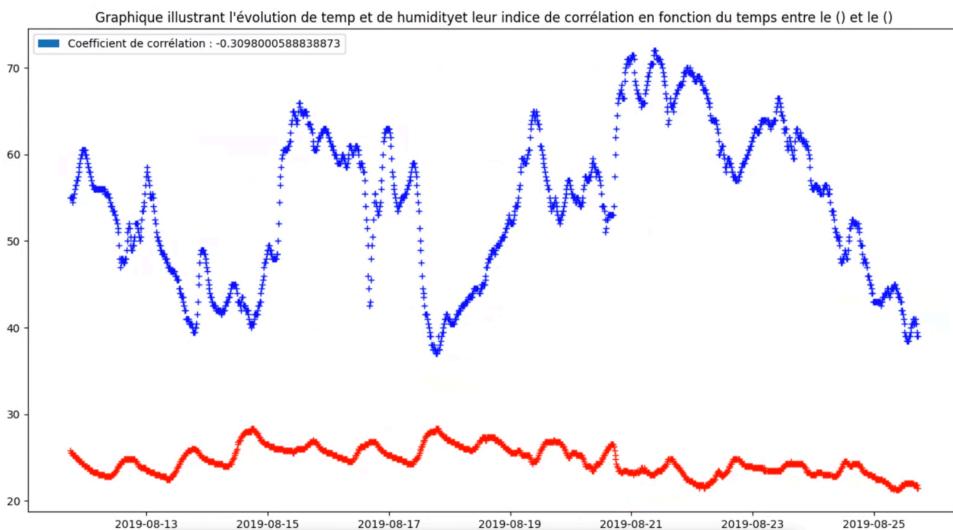
- **dtw_method(s,t,fen)** : on quantifie la distance entre les deux courbes
- **prim_com(data, cat, sen1, sen2, *btemp)**: La liste des différences est retournée. On compare terme à terme et non par rapport au temps.
- **sim_prim_draw(data, cat, sen1, sen2, *args)** : On trace.
- **comparaison_primaire(e, data, cat, sen1, sen2, *args)** : On va définir un seuil qui va permettre de statuer la différence, c'est-à-dire, si la différence est inférieur à ce seuil alors on peut qualifier de semblable les valeurs correspondant à cette différence.

c) Affichage



Note : capteurs 1 et 2 pour co2, affichage du DTW

VII. Bonus 1



VIII. Bonus 2

Voir programme.

IX. Conclusion

Ce projet nous a permis de développer nos capacités de travail à distance grâce à l'outil GitHub. De plus, approfondir nos connaissances sur l'utilisation des paramètres statistiques met en évidence la nécessité de pouvoir comparer des séries temporelles pour avoir une idée du comportement énergétique d'un bâtiment.

Lien : https://github.com/lgmaureen23/python_rendu

Nom du rapport : Rapport_python_rendu_FRIDHI_Ilies_LEGER_Maureen.docx

Nom du programme Python : READ_CSV_FINAL_FRIDHI_Ilies_LEGER_Maureen.py