RAPPORT DE PROJET

***Groupe-30***

PROJET DATA

*Projet réalisé par :*

Hicham MATRAB

Yacine HEBBOUDJI

*Projet encadré par :*

M. N. LAGA

M. P.E-MASSE

Table des matières

[I. Introduction 3](bookmark://_Toc57663612#_Toc57663612)

[a. Contexte 3](bookmark://_Toc57663613#_Toc57663613)

[b. Présentation du sujet 3](bookmark://_Toc57663614#_Toc57663614)

[II. Fonctionnalités 4](bookmark://_Toc57663615#_Toc57663615)

[a. Affichage de courbes montrant l’évolution d’une variable sur un intervalle prédéfinie. 4](bookmark://_Toc57663616#_Toc57663616)

[b. Affichage des valeurs statistiques 7](bookmark://_Toc57663617#_Toc57663617)

[c. Calcul l’indice humidex 8](bookmark://_Toc57663618#_Toc57663618)

[d. Calcul l’indice de corrélation 8](bookmark://_Toc57663619#_Toc57663619)

[III. Analyse Sujet : Sujet A & C 10](bookmark://_Toc57663620#_Toc57663620)

[IV. Utilisation de Github 13](bookmark://_Toc57663621#_Toc57663621)

# Introduction

## Contexte

Ce projet, à but « universitaire », a pour ambition de nous faire apprendre le design d’algorithme. Il a aussi pour vocation de nous faire maîtriser les outils de travail en équipe tels que GitHub.

Pour ce faire, il nous ait fourni un fichier au format CSV dans lequel se trouve un jeu de données à étudier, issu d’une campagne de mesure au sein d’un bâtiment de bureau.

Ce jeu de données, proposé par la start-up Kandu, contient cinq données relatives au bâtiment de bureau, prises par des capteurs environ toutes les 15 minutes, du ‘11 août 2019 à 11h30m50’ au ‘25 août 2019 à 17h47m8s’.

On recense une mesure de la température ambiante (°C, < type ‘float’>), de l’humidité relative (%, < type ‘float’>), du niveau sonore (dBA, <type ‘float’>), du niveau lumineux (lux, <type ‘int’>), de la quantité de CO2 (ppm, <type ‘int’>).

À cela s’ajoutent deux autres attributs : une relative à l’échéance de la mesure prise et l’autre qui correspond au numéro du capteur.

## Présentation du sujet

Il s’agit ici de réaliser un programme sous forme d’un script avec arguments, qui ne plantent pas facilement (try catch), qui devra lire les données du fichier CSV, accepter les arguments d’entrées de la ligne de commande et générer en sortie les résultats attendus.

Il nous ait aussi demandé en autres de fournir quatre fonctionnalités au script :

* Afficher des courbes montrant l’évolution d’une variable en fonction du temps, avec éventuellement la possibilité de spécifier un intervalle de temps dans la ligne de commande ;
* Afficher les valeurs statistiques sur la courbe ;
* Calculer l’indice ‘Humidex’ ;
* Calculer l’indice de corrélation.

Le sujet proposé aux groupes A&C est le suivant :

*Trouvez- vous des anomalies dans les données, que pouvez-vous en conclure ?*

*Proposez et implémentez un algorithme permettant de relever les anomalies automatiquement et de les montrer sur les courbes.*

# Fonctionnalités

La première étape consistait à lire le fichier CSV fourni. Pour cela, on a tenté une première approche ‘naïve’ en faisant appel à la fonction open pour l’ouverture du fichier et ‘reader.csv’ pour le lire et pour ensuite extraire les lignes une à une avec ‘row’ (cf. code #1)

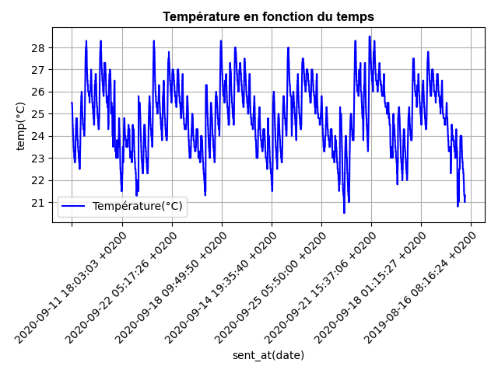
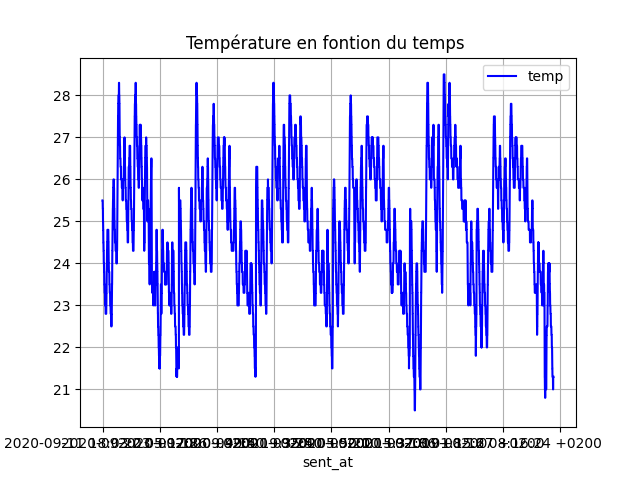
La première difficulté rencontrée a été un problème au niveau de la chaine de caractère. En effet, notre chemin d’accès comportant des slash \ , pour pouvoir les garder tels quels, il fallait utiliser à la place des doubles slash \\ ou bien préfixer le premier guillemet par un ‘r’ pour obtenir une chaîne brute ( « raw string »).

Cependant, la librairie pandas étant très performante pour manipuler des données d’un fichier CSV, nous avons finalement préférer poursuivre notre travail avec pour manipuler nos données. (cf. code 2.2)

## 

## Affichage de courbes montrant l’évolution d’une variable sur un intervalle prédéfinie.

Voici l’affichage de la courbe montrant l’évolution de la température au cours du temps :

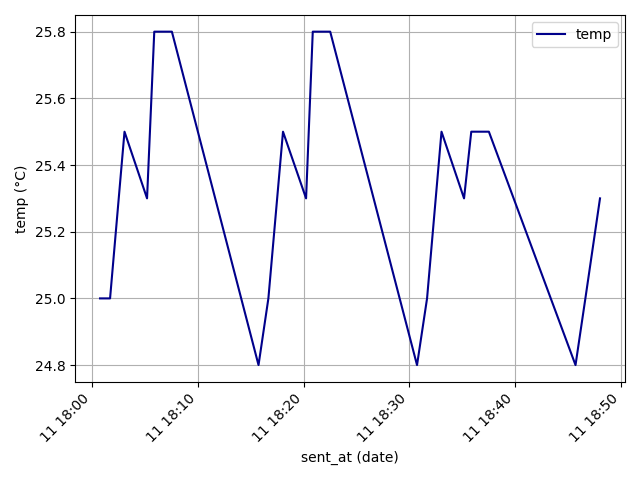


**Figure 2.a.1 : Evolution de la température en fonction du temps**

**au sein du bâtiment de bureau**

Le graphique de droite par rapport à celui de gauche permet de montrer l’évolution de notre travail dans l’amélioration du graphe.

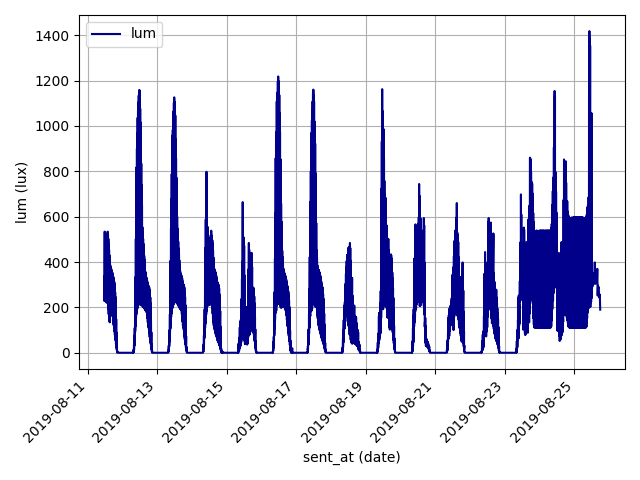
En y spécifiant l’intervalle de temps ; 2019-08-11 18:00:44+02:00 à 2019-08-11 18:48:03+02:00 :



**Figure 2.a.2 - Evolution de la température au cours du temps au**

**sein du bâtiment de bureau (avec intervalle prédéfinie)**

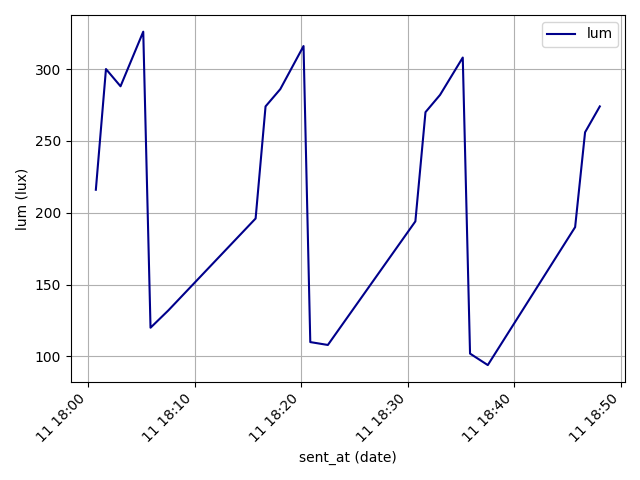
Voici l’affichage de la courbe montrant l’évolution de la lumière au cours du temps :



**Figure 2.a.3 - Evolution de la lumière au cours du temps au**

**sein du bâtiment de bureau**

En y spécifiant le même sous intervalle de temps que pour la variable précédente :



**Figure 2.a.4 - Evolution de la lumière au cours du temps au**

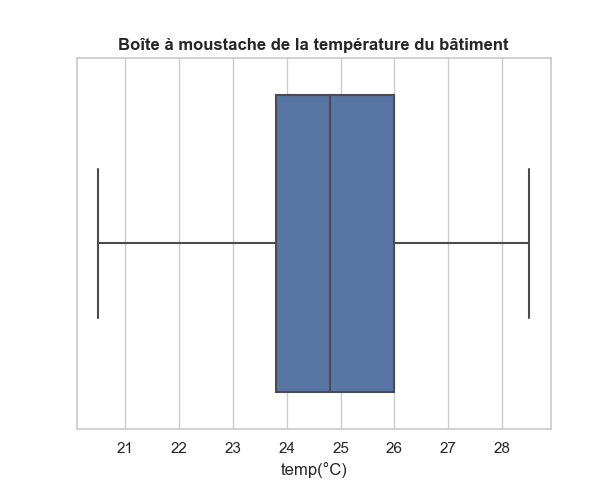
**Sein du bâtiment de bureau**

## Affichage des valeurs statistiques

Méthode choisie : utilisation d’une boîte à moustache

Nous avons choisie d’utiliser une boite à moustache qui rassemble toutes les données statistiques voulues : maximum, minimum, premier/troisième quartile, médiane sont tous regroupés sur un même graphe.

Pour la variance, nous pensons qu’elle peut aussi être vu par le diagramme mais nous ne parvenu à savoir comment.



maximum

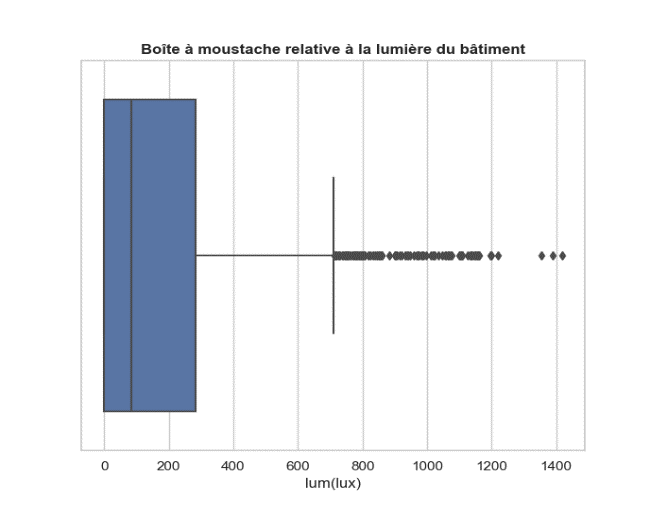
minimum

Médiane

**Figure 2.b.1- Diagramme en boîte synthétisant les données statistiques**

**canoniques de la courbe représentée par la *Figure 2.a.1*.**

De même, voici les valeurs statistiques pour la lumière :

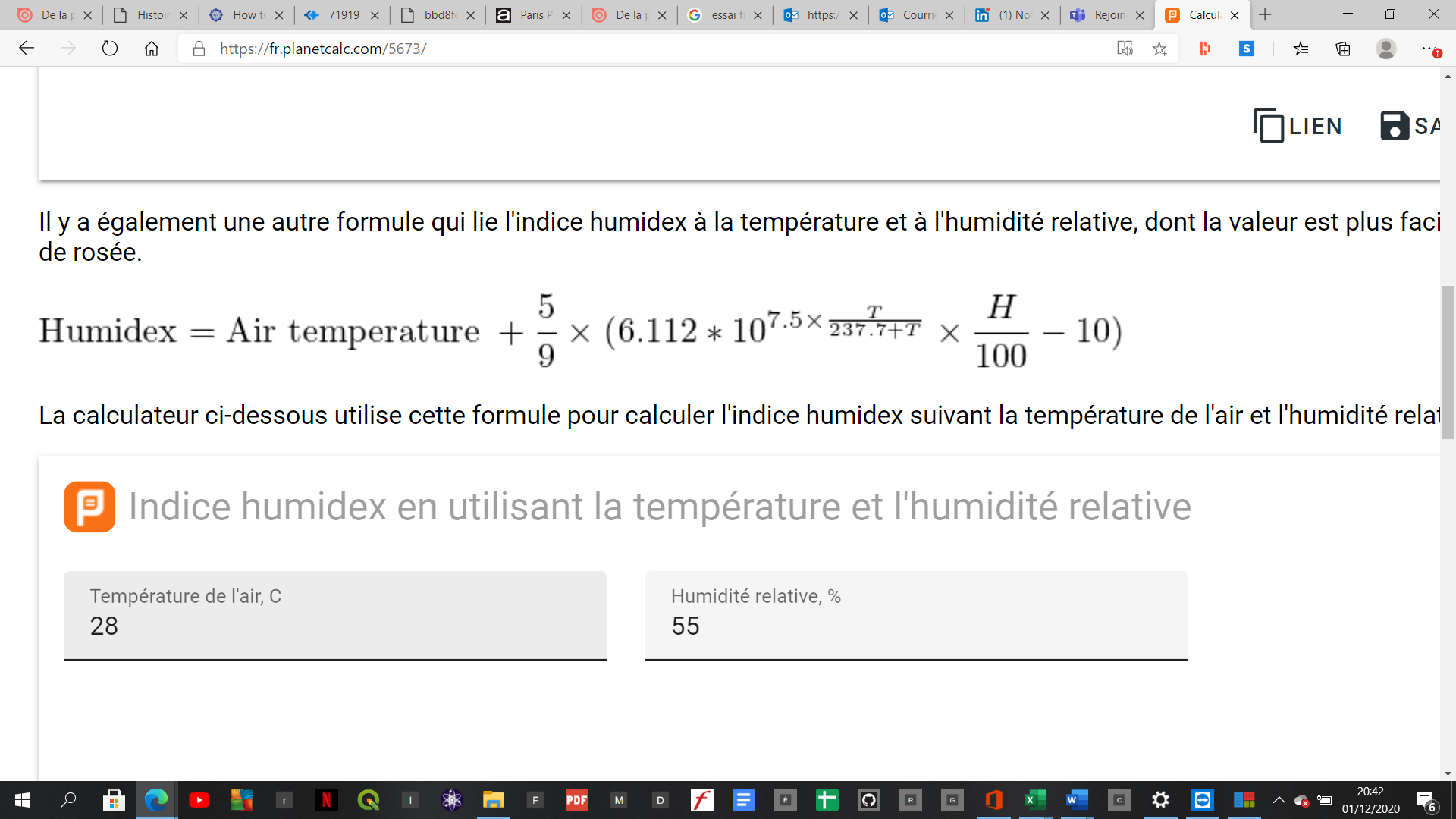


**Figure 2.b.1- Diagramme en boîte synthétisant les données statistiquescanoniques de la courbe représentée par la *Figure 2.a.3*.**

## Calcul l’indice humidex

Méthode choisie : renvoie de la liste des valeurs de l’humidex à partir d’un algorithme calculant cet indice pour chaque valeur du taux d’humidité.

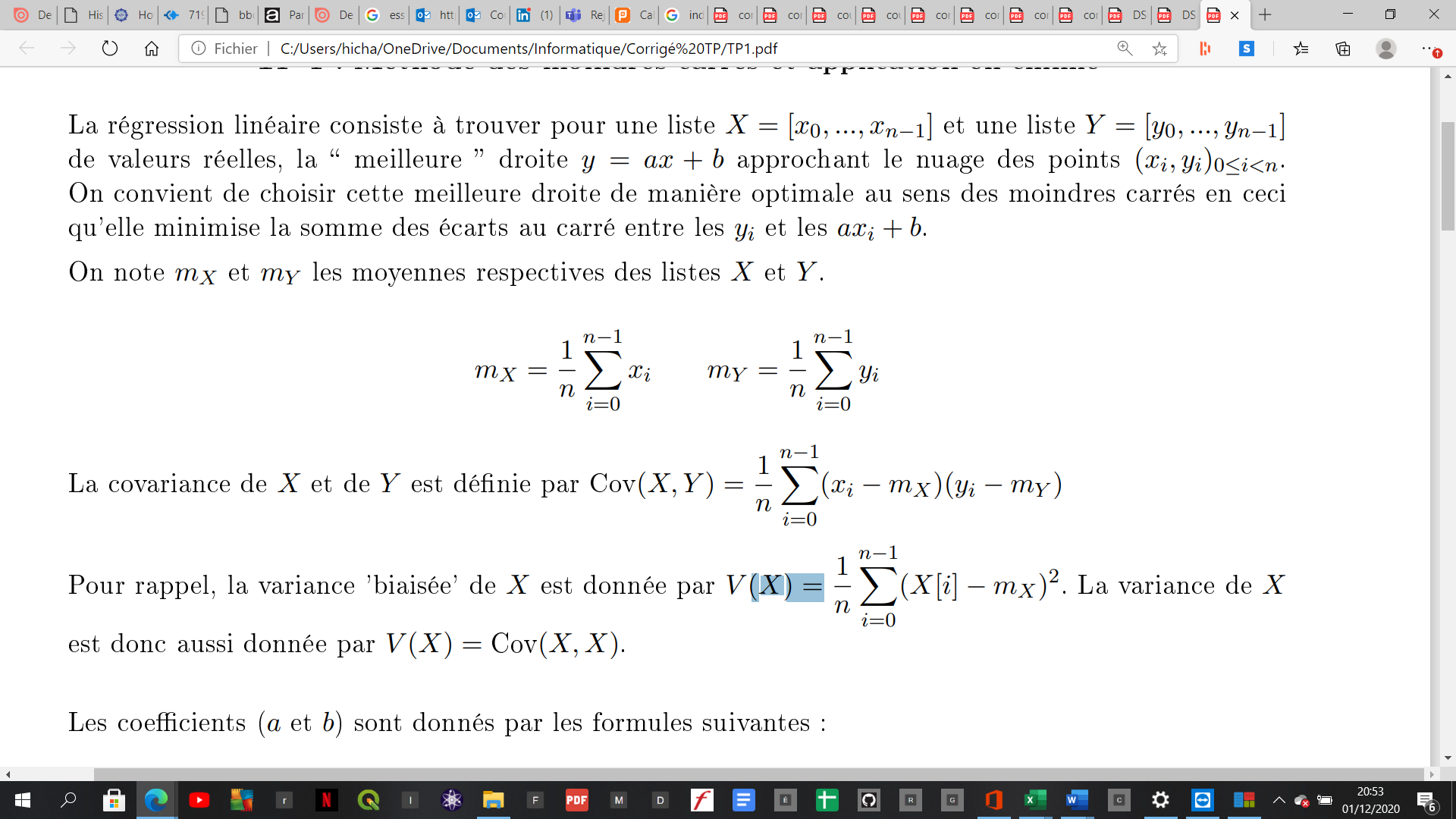
Nous avons pu calculer la valeur de l’indice humidex avec la formule reliant la température ambiante et l’humidité relative toutes les deux présentes dans les données du fichier CSV :

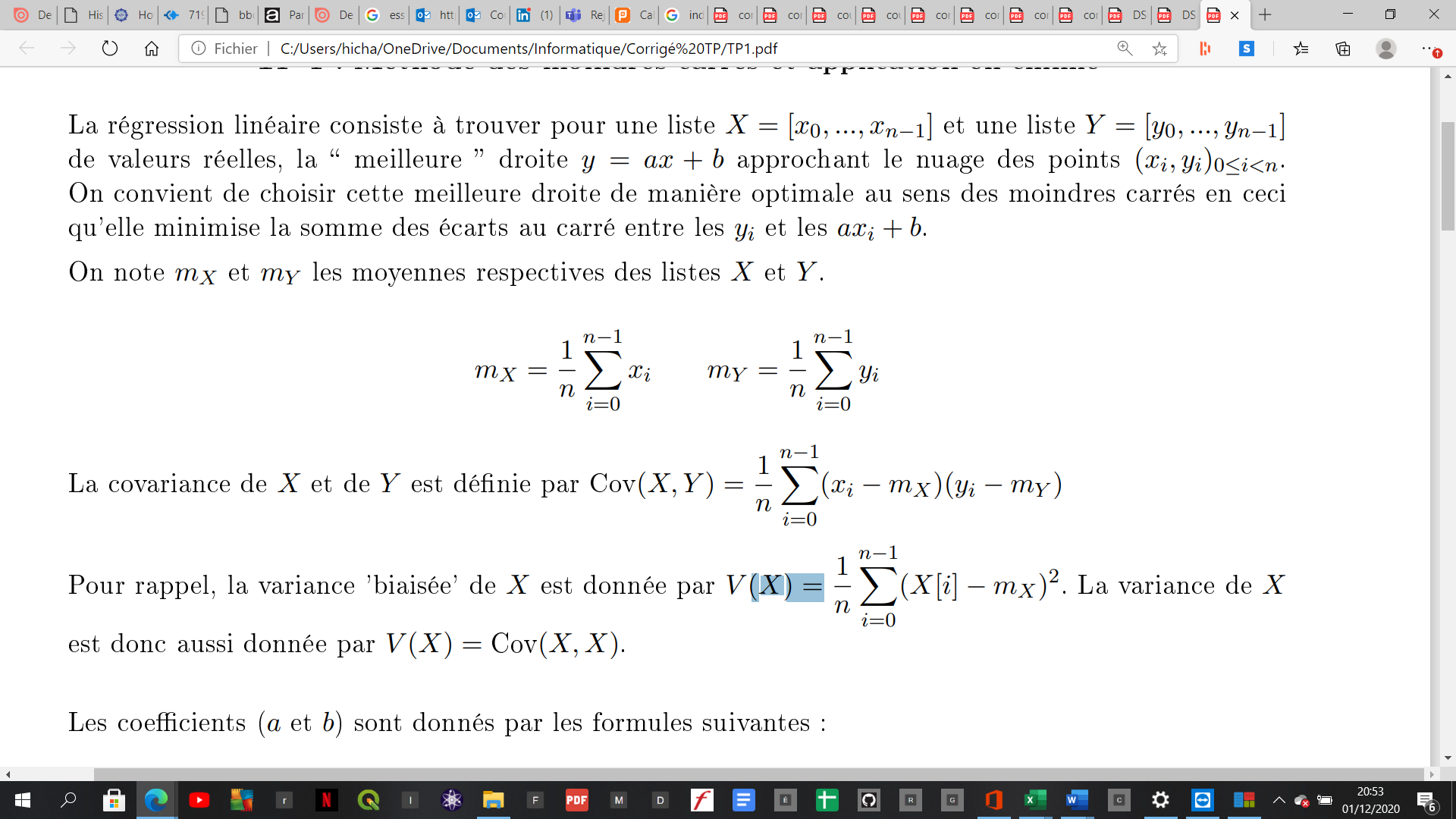


Le calcul de cet indice s’agissant d’un simple calcul algébrique, la fonction associée au calcul ( code ##5) a une complexité constante : O(1).La complexité est inchangée lorsque on retourne la valeur de l’humidex avec nos données grâce à la libraire pandas.

## Calcul l’indice de corrélation

Par définition, en introduisant deux variables aléatoires réelles X et Y, l’indice de corrélation vaut :

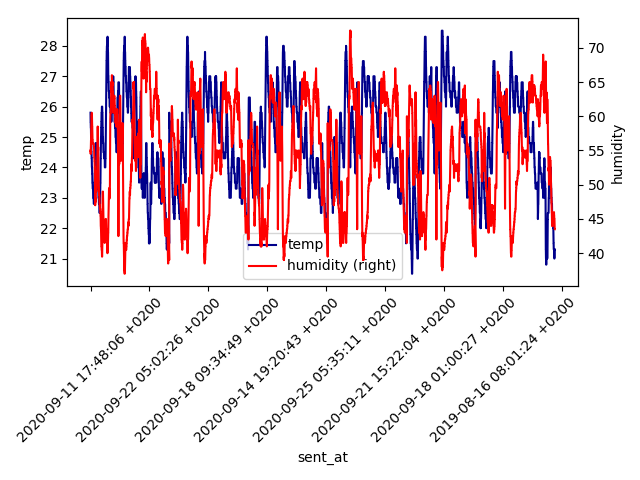




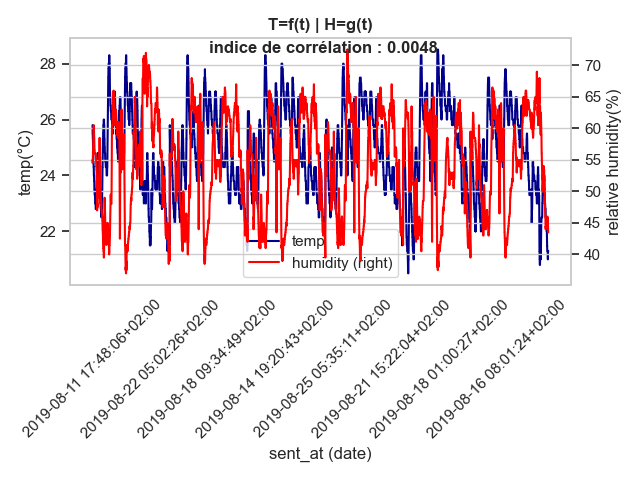
Avec :

; désignant respectivement la moyenne de variable aléatoire X={x1 ,… ,xn celle de Y ; les

Dans un premier temps, on peut implémenter une fonction moyenne qui calcule la moyenne d’une liste. ( cf. code ##4). La complextié de cet algorithme est : O (len(L)) c’est-à-dire de la taille de la liste appelant L la liste en question. Ainsi , la complexité est linéaire : O (n) avec n la longueur de la liste L. A partir de la définition de la covariance, on peut implémenter une fonction covariance, qui prend en arguments deux listes et qui calcule la covariance des ceux deux listes. La complexité de l’algorithme sera en O(n) avec la longueur de l’une des deux listes (supposées de même longueur), d’après le ##code4 la fonction possède une boucle for qui parcours n éléments, et les calculs au sein de la boucle sont en O(1).



**Figure 2.c.1– Evolution de la température et de l’humidité au cours du temps**



**Figure 2.c.1– Evolution de la température et de l’humidité au cours du temps**

**(avec indice de corrélation)**

# Analyse Sujet : Sujet A & C

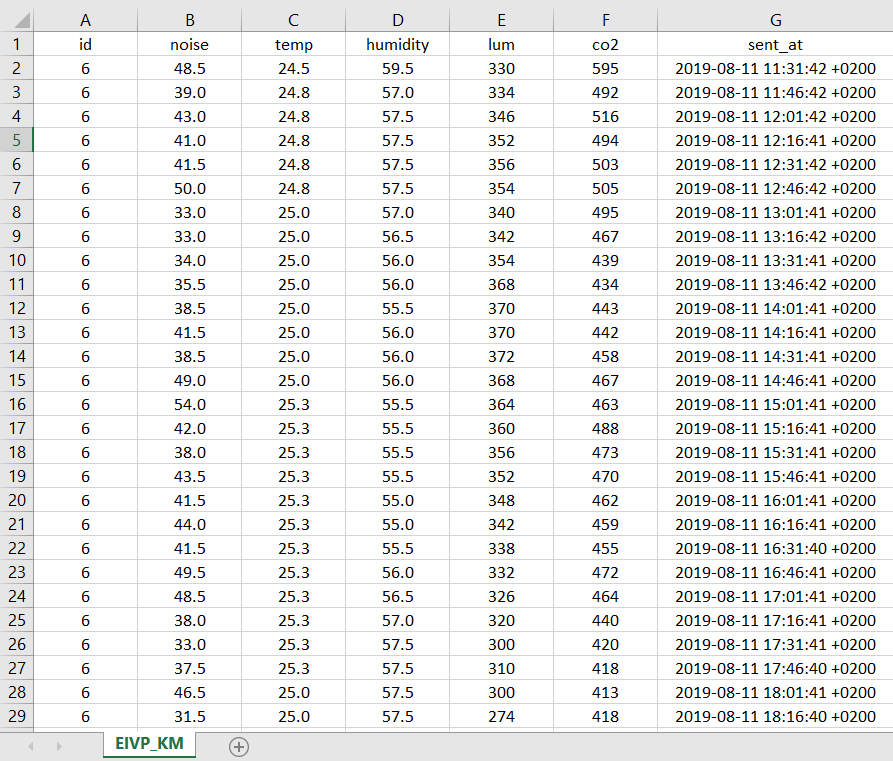
Le sujet était le suivant :

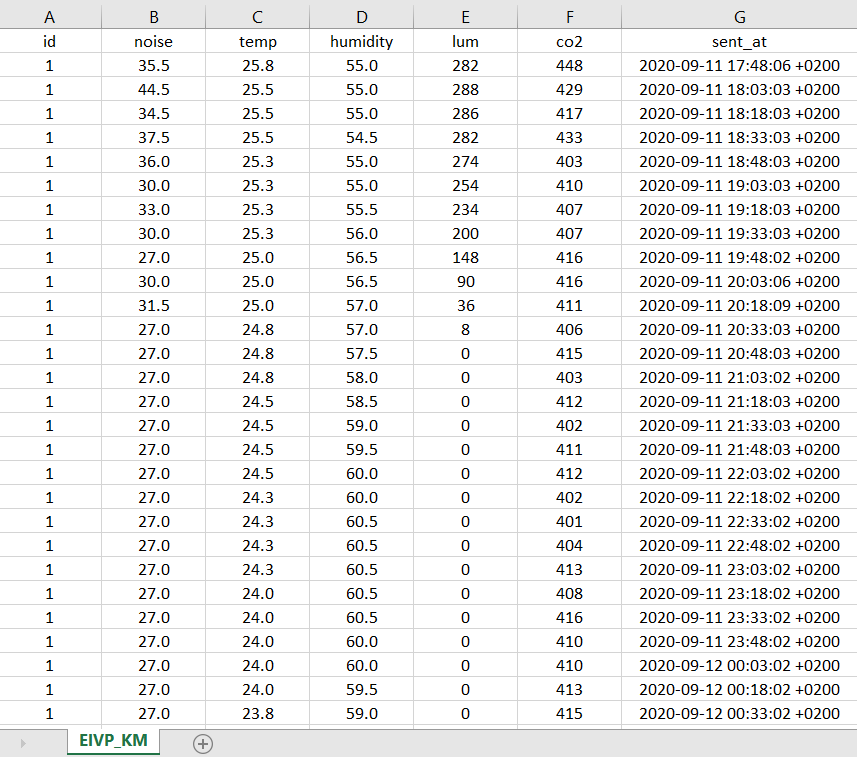
*Trouvez- vous des anomalies dans les données, que pouvez-vous en conclure ?*

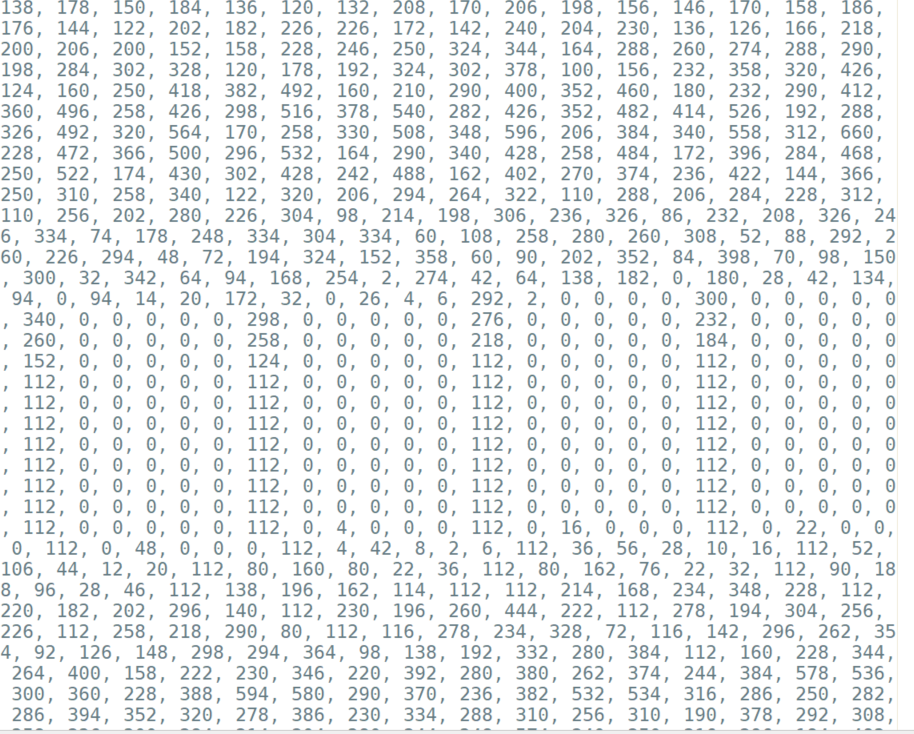
*Proposez et implémentez un algorithme permettant de relever les anomalies automatiquement et de les montrer sur les courbes.*

Dans un premier temps, nous avons cherché les anomalies manuellement : cela nous a permis d’élaborer une stratégie en fonction des anomalies détectées. Les anomalies que nous avons relevées et corrigées via le programme “anomalie” (qui renvoie les listes des valeurs du bruit, de la température, de l’humidité, de la lumière et du CO2 dans le bon ordre chronologique) sont les suivantes :

* Le capteur 6 n’a pas été programmé avec la bonne date (décalage de près d’un an)



* Le capteur 1 n’a pas été programmé avec la bonne heure (démarre plus tard et s’arrête plus tard) 
* Le capteur 1 n’a pas enregistré à des intervalles exactement régulier (anomalie non corrigeable puisqu’on ne peut pas modifier le temps)
* Le capteur 5 n’a pas autant de valeurs que les autres capteurs
* Le capteur 5, sur une cinquantaine de valeurs de CO2 bug et n’affiche que la valeur « 112 » au lieu du 0 des autres capteurs (seule anomalie que nous n’avons pas réussi à corriger via le programme “anomalie”)



Pour corriger les anomalies, on part du principe que les anomalies sont les mêmes dans un même capteur. Donc on enregistre dans un premier temps chaque donnée (bruit, température, humidité, luminosité et quantité de CO2) dans des listes distinctes à la fois par donnée et par capteur. Ensuite, en fonction du nombre de données récoltées par chaque capteur, on a créé des boucles qui permettent de stocker dans une liste "bruit" par exemple toutes les données de tous les capteurs dans le bon ordre chronologique.

:

# Utilisation de Github

GitHub, à première vue, est un outil très complexe d’utilisation. En effet, rares sont les personnes qui ont déjà manipulé le terminal de leur PC avant GitHub. L’enjeu ici a été de comprendre et apprendre à utiliser l’interface du terminal pour pouvoir « push » et « pull » le travail effectué de part et d’autre.

Nous avons rencontré un problème pour la manipulation du fichier, nous avons dû recommencer et tout push à la dernière minute pour que les avancées du projet soient visibles.

Concrètement, nous avons appris à inclure un fichier sur le fichier git, à le modifier. Une erreur à ne surtout pas faire, qui a causé quelques soucis dans notre cas : ne surtout pas « push » un document en même temps pour éviter que l’ordinateur ne refuse les deux modifications.

# Conclusion

Pour ce qui est du code en lui-même, nous avons pu, grâce aux différents codes (code général et fonction anomalie) analyser les données des capteurs et en corriger les anomalies. En effet, les anomalies citées plus haut ont été corrigées et cela nous a permis d’obtenir des courbes plus cohérentes et “lissées” par rapport à ce que l’on a obtenu initialement.

Ce projet, dans un premier temps, nous a paru relativement compliqué. En effet, mis à part la maîtrise du langage Python sur une interface semblable à pyzo, les connaissances acquises en classe préparatoire ne nous ont pas aidé dans l’utilisation du terminal de l’ordinateur, essentielle dans ce projet, que ce soit pour l’utilisation de GitHub ou pour lancer le code.

Au-delà de ça, le projet nous a fait acquérir une réelle expérience de groupe et des bases solides quant à l’utilisation d’outils de programmation.