Rapport projet programmation : Sujet 2

Méthode de travail et utilisation de GitHub :

La majeure partie du projet s’étant faite à distance, nous avons décidé de travailler simultanément, et non chacun de notre côté, sur les programmes. En effet, par session de 2 à 4h nous nous appelions, Quentin faisait un partage d’écran et nous réfléchissions ensemble. Puis il enregistrait les modifications sur GitHub.

Programmes :

Pour commencer on a importé les modules maths, matplotlib, numpy et datetime.

La première fonction tab\_capteurs() permet de transformer le tableau excel en listes de liste. Pour cela on a créé six listes correspondant aux six capteurs dans lesquelles on a mis les valeurs pour les différentes mesures. On a aussi ignoré la première ligne du tableau avec next() et transformé les valeurs en int et float.

Puis on a implémenté une fonction valeurs\_stats(liste) qui prend en argument une liste et affiche son minimum, maximum, la moyenne, l’écart-type et la médiane. La moyenne et l’écart-type ont été arrondies à deux chiffres après la virgule avec la fonction round. Pour la médiane on avait préalablement implémenté une fonction de tri rapide nommée tri(liste). N’ayant pas utilisé panda, ces valeurs sont calculées manuellement. On obtient ainsi les valeurs statistiques pour chaque grandeur de chaque capteur. Pour vérifier le programme, on l’a essayé sur des petites listes dont on connaît les valeurs statistiques.

La fonction courbe\_simple(liste1,liste2,a,b,c) prend en argument deux listes, le nom du graphique, de l'abscisse et de l’ordonnée et affiche la courbe de la grandeur en fonction du temps. Elle affiche en plus sur le graphe les valeurs statistiques grâce à la fonction text() qui permet aussi de choisir l’emplacement du texte.

La fonction test() permet de vérifier la fonction précédente appliquée aux valeurs de l’Excel, donc avec les listes de tab\_capteurs. On l’a juste vérifié pour la liste des températures mesurées du capteur 1 et cela fonctionne.

La fonction corrélation(L,P) a comme arguments 2 listes, et donne la corrélation entre ces deux listes, c’est-à-dire à quel point ces 2 listes se “ressemblent”. La fonction graphe\_corrélation affiche les deux courbes et leur corrélation.

On a par la suite créé la fonction humidex calculant l’indice humidex. En argument il y a une liste de températures et une liste d'humidité. Dans une nouvelle liste on a ajouté les valeurs humidex qu’on a calculé avec la formule :

Humidex=T+(5/9)\*(6.112\*107,5\*(T/237,7+T) \*(H/100)-10)) avec T pour température et H pour l’humidité. La fonction renvoie une liste de valeurs humidex.

Pour pouvoir comparer les dates on les a transformées en format comparable avec la fonction dt, et les fonctions limitation\_temps et limitations\_temps\_totale permettent de garder seulement les valeurs qui se trouvent dans l’intervalle de temps qui nous intéresse pour chaque capteur.

Afin de vérifier la fonction graphe\_correlation et dt avec les valeurs du csv, on a implémenté une fonction test2 qui affiche le graphe et la corrélation entre la température de 2 capteurs. Cela fonctionne, on peut donc commencer à mesurer les similarités pour chaque capteur et chaque dimension. Pour cela on a écrit la fonction graph, affichant pour une dimension les six courbes (une pour chaque capteur) qu’on a décliné pour toutes les dimensions avec la fonction finale mesure\_similarité. On a auparavant créé des listes avec les valeurs de chaque dimension de chaque capteur, par exemple la liste B (bruit) regroupe les valeurs du bruit pour chaque capteur sous forme de sous liste. Cela affiche alors les graphes des similarités et les corrélations entre les capteurs.

Résultats :

-certaines corrélations étaient supérieures à 1 ce qui devrait être impossible, on a donc cherché où était notre erreur. Finalement le problème venait des chiffres significatifs des écarts-types et espérances dans la fonction valeurs\_stats du début. Nous avions arrondi à un chiffre derrière la virgule, ce qui était bien trop grossier. Nous avons donc modifié la fonction corrélation pour ne plus avoir besoin d’appeler la fonction valeurs\_stats.

-la corrélation est de 1 pour les capteurs 1 et 6, ceci quelle que soit la dimension. Leurs courbes sont confondues. On peut en déduire que les capteurs 1 et 6 sont placés au même endroit, ou assez proches l’un de l’autre.

-pour le niveau sonore on remarque que les capteurs 3 et 5 enregistrent le plus de “bruit” tandis que les capteurs 2 et 4 en captent moins.

-le niveau de luminosité est bien plus élevé au capteur 3 que chez les autres capteurs. Puis vient le capteur 4, le 2, le 6 (et donc le 1) et enfin le 5. On en déduit que le capteur 3 se situe près d’une fenêtre assez exposée à la lumière du jour. Pareil pour le capteur 4, mais moins exposé à la lumière du jour. Les autres capteurs sont moins exposés à la lumière.

-la température du capteur 3 est légèrement supérieur aux autres, ce qui peut être rapproché du niveau de luminosité élevé.

-la quantité de CO2 est élevée et similaire au niveau des capteurs 2 et 4. Les capteurs 1,3,5 et 6 enregistrent un taux de CO2 moins important mais comparable, avec un taux plus faible pour le 5.

Bonus :

Les courbes du niveau sonore nous renseignent sur les heures d’occupation des bureaux. On voit bien sur cinq jours d’affilés un niveau sonore élevé et similaire puis deux jours où le niveau est bien plus bas.

Graphes obtenus :









