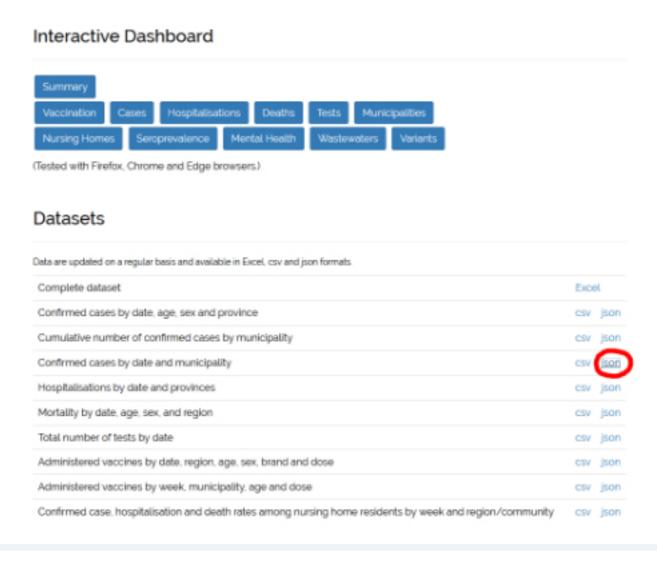
prendre le fichier : 'Confirmed cases by date and municipality' en format 'json'



communes. On choisi ce format car c'est un format facilement lisible par un programme informatique sans devoir formatter et convertir un fichier excel par exemple. De plus ce fichier est plus léger que le classeur de base excel proposé par sciensano. Ce fichier porte le nom de : COVID19BE_CASES-MUNI.json

Ce fichier contient les données concernant le nombre de cas confirmés de contamination à la Covid19 par

(*NISS': '62011', 'DATE': '2021-02-27', 'TX DESCR NL': 'Bitsingen', 'TX DESCR FR': 'Bassenge', 'TX ADM DSTR DESCR FR': 'Arrondissement de Liège', 'PROVINCE': 'Liège', 'REGION': 'Wallonia', 'CASES': '<5'

Le fichier est formatté comme ceci : C'est une liste de dictionnaires dans ces dictionnaires ont sait accèder aux valeurs grâce à ces clés : **DATE** TX_DESCR_NL TX_DESCR_FR TX_ADM_DSTR_DESCR_NL TX_ADM_DSTR_DESCR_FR **PROVINCE** REGION **CASES**

Taille

Type

Les clés qui nous interessent ici :

TX_DESCR_FR qui correspond au nom des communes ciblées par le projet (Bruxelles, Schaerbeek, Evere, Wo-luwe-Saint-Lambert, Etterbeek), La DATE et enfin CASES qui correspond au nombre de cas confirmés.



Les bouts de code en

pour fonctionner (lik

Le script Python contenu dans le dossier '1. Conversion données de Sciensano' crée un deuxième fichier appelé 'COVID_5BXL.json' dans lequel les informations voulues ont été triées et reconverties. :udes > Iviath 3eme > 1. Conversion données de Sciensano

autres.py COVID_5BXL.json	08-06-22 15:06 08-06-22 15:08	Fichier PY Fichier JSON	1 Ko 85 Ko
COVID_3BXE.json COVID_19BE_CASES_MUNI.json	24-04-22 11:16	Fichier JSON	71.445 Ko
Lien vers base de données.txt	24-04-22 11:05	Document texte	1 Ko
quel fichier prendre.PNG	24-04-22 11:24	Fichier PNG	97 Ko
script.py	24-04-22 16:23	Fichier PY	4 Ko

Modifié le

tions 'pip' afin des les installer automatiquement (ou met à jour si nécessaire).

Une fois le fichier 'COVID_5BXL.json' créé il suffit de le copier dans le dossier '4.0.1 Ecart type Iordan

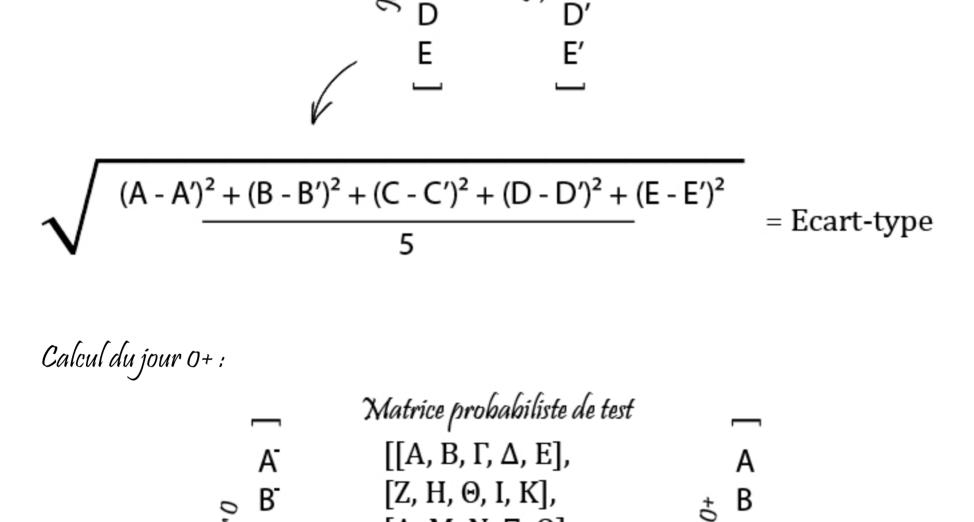
Jour 0

2. La méthode des écarts-type et des oscillateurs

Nom

(not neg)'. Il existe d'autre dossier avant ça, mais comme vous vous doutez c'est dans ces dossiers que les autres parties ont été faites/testées... L'idée principale du projet est de générer un grand nombre de matrice probabiliste de test et de pouvoir comparer le résultat entre un vecteur de base composé des 5 communes choisies et le produit de ces matrices par

rapport au vecteur du jour suivant. La formule de l'écart-type permet de déterminer quelle matrice probabiliste est la plus 'proche' de la réalité (résultat de l'ecart-type le plus proche de 0) Ecart Type:



* [Λ, M, N, Ξ, O],

 $[\Pi, P, \Sigma, T, \Upsilon],$

 $[\Phi, X, \Psi, \Omega, \tilde{I}]]$

Concernant les matrices de bases, qui constituent le point de départ des calculs, il a été décidé de partir de matrices

probabilistes grossières constituées soit de 0.5 soit de 0. Afin de limiter les calculs dès le départ.

qui contient lui meme toutes les combinaisons possible de matrices probabilistes possible avec les termes décrit juste avant. Une petite exception, il existe dans ce fichier une matrice constituée exclusivement de 0, car au vu des données issues de sciensano il apparait que certains jours ne contiennent aucun cas de Covid dans les communes choisies... Cette matrice à tout naturellement un écart-type de 0 dans ces situations.

Dans le dossier, il existe un fichier 'creation des matrices de transition de base.py' qui s'occupe de créer le fichier 'mtx_base'

Les oscillateurs :

matrices ayant un lien avec la première tout en fournissant la possibilité d'affiner la précision de celles-ci. Ce qu'on a appelé 'zoom' entre nous, car on part d'une matrice très grossière pour affiner/zoomer petit à petit vers un résultat idéal. C'est ici que j'ai eu l'idée d'un oscillateur, qui est une fonction écrite en .py et lisible dans les sous dossiers 'rsc'. Ces oscillateurs, oscillent autour des valeurs trouvées à une étape précédentes pour créer de nouvelles valeurs. Les oscillateur fonctionne <u>par ligne,</u> et s'appliquent sur un ordre de grandeur et non de position sur les matrices. Par exemple si on prend l'oscillateur [-1, -1, 0, 1, 1] on remarque que la somme de ses

En partant des matrices probabilistes de base choisie à l'étape précédente il faut désormait générer une série d'autres

restent à 1. Cet oscillateur va diminuer de 1 la valeur la plus élevée et sa suivante de la matrice probabiliste retenue, ne rien faire pour le terme suivant et augmenter de 1 les deux termes plus faibles. $[0.0, 0.5, 0.0, 0.5, 0.0] \xrightarrow{Osc, divide=10} [-1, -1, 0, 1, 1] = [0.1, 0.4, 0.1, 0.4, 0.0]$ $[0.1, 0.4, 0.1, 0.4, 0.0] \xrightarrow{Osc, divide=100} [1, -1, 0, -1, 1] = [0.09, 0.41, 0.1, 0.39, 0.01]$

termes est égale à 0 car les oscillateurs opère par addition/soustrac-

tion, de ce fait la somme des termes des matrices probabilistes

C'est ainsi que l'on génère des matrices probabilistes secondaires. Comme les oscillateurs fonctionnent par lignes et qu'il y a 5 lignes dans notre énoncé les oscillateurs génèrent : le nombre d'oscillateurs exposant 5. Ici il y a 12 oscillateurs qui génèrent au total 248832 matrices probabilistes de test à chaque passe d'oscillateur. Lors de ces passes ont recalcul à chaque fois l'écart type pour savoir laquelles de ces matrices est la meilleure base pour repartir sur une passe suivante... jusqu'a la précision voulue pour l'écart-type ou bien lorsque le nombre

de passe maximal à été atteint. (il est important de définir un nombre maximum de passe car on risque de tomber dans une boucle infinie...)

Ces scripts sont identiques, a la seule différence que P1 s'occupent du premier quart des données, P2 du deuxième quart

et ainsi de suite... Car comme il y a 775 passages de jour multiplié par 248832 matrices à chaque fois.. Ca fait lourd

pour un seul processus et demande du temps calcul... Chacun de ces scripts génèrent à leur tour un fichier de résultat : 'result_MTX_PX.json'

Ainsi dans le dossier il y a P1, P2, P3 et P4.

Et c'est dans ces fichiers que se trouvent les résultats de toutes les matrices retenues ainsi que l'écart-type calculé à chaque fois... Ce sont des fichiers .json, qui sont représentent une liste de dictionnaires dont la clé est la date du jour, et dans laquelle la chaine 'r' contient le résultat de l'écart-type et la chaine 'mtx' la

Comme les oscillateurs génèrent un grand nombre de matrices, il a été décidé de splitter les calcules en 4.

Stéphane.