

1. conversion des données de Sciensano

Depuis le site : <https://epistat.wiv-isp.be/covid/>

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

Interactive Dashboard	
<div>Summary</div> <div>VaccinationsCasesHospitalisationsDeathsTestsMunicipalities</div> <div>Nursing HomesSeroprevalenceMental HealthWastewaterPatients</div> <div>(Tested with Firefox, Chrome and Edge browsers.)</div>	
Datasets	
Data are updated on a regular basis and available in Excel, csv and json formats.	
Complete dataset	Excel
Confirmed cases by date, age, sex and province	csv json
Cumulative number of confirmed cases by municipality	csv json
Confirmed cases by date and municipality	csv json
Hospitalisations by date and provinces	csv json
Mortality by date, age, sex, and region	csv json
Total number of tests by date	csv json
Administered vaccines by date, region, age, sex, brand and dose	csv json
Administered vaccines by week, municipality, age and dose	csv json
Confirmed case, hospitalisation and death rates among nursing home residents by week and region/community	csv json

Ce fichier contient les données concernant le nombre de cas confirmés de contamination à la Covid19 par communes. On choisi ce format car c'est un format facilement lisible par un programme informatique sans devoir formater 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

["NIS5": "11002", "DATE": "2020-09-15", "TX\_DESCR\_NL": "Antwerpen", "TX\_DESCR\_FR": "Anvers", "TX\_ADM\_DSTR\_DESCR\_NL": "Arrondissement Antwerpen", "TX\_ADM\_DSTR\_DESCR\_FR": "Arrondissement d'Anvers", "PROVINCE": "Antwerpen", "REGION": "Flanders", "CASES": 134]  
["NIS5": "44020", "DATE": "2021-04-01", "TX\_DESCR\_NL": "Gavere", "TX\_DESCR\_FR": "Gavere", "TX\_ADM\_DSTR\_DESCR\_NL": "Arrondissement Gent", "TX\_ADM\_DSTR\_DESCR\_FR": "Arrondissement de Gand", "PROVINCE": "Oost-Vlaanderen", "REGION": "Flanders", "CASES": 9]  
["NIS5": "62011", "DATE": "2021-02-27", "TX\_DESCR\_NL": "Bilsingen", "TX\_DESCR\_FR": "Bassenge", "TX\_ADM\_DSTR\_DESCR\_NL": "Arrondissement Luik", "TX\_ADM\_DSTR\_DESCR\_FR": "Arrondissement de Liège", "PROVINCE": "Wallonie", "REGION": "Wallonia", "CASES": 45]

Le fichier est formatté comme ceci :  
C'est une liste de dictionnaires dans ces dictionnaires  
ont soit accéder aux valeurs grâce à ces clés :

NIS5  
DATE  
TX\_DESCR\_NL  
TX\_DESCR\_FR  
TX\_ADM\_DSTR\_DESCR\_NL  
TX\_ADM\_DSTR\_DESCR\_FR  
PROVINCE  
REGION  
CASES

Les clés qui nous interessent ici :

TX\_DESCR\_FR qui correspond au nom des communes ciblées par le projet (Bruxelles, Schaerbeek, Evere, Woluwe-Saint-Lambert, Etterbeek), La DATE et enfin CASES qui correspond au nombre de cas confirmés.

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.

nom	Modifié le	Type	Taille
autres.py	08-06-22 15:06	Fichier PY	1 Ko
COVID_5BXL.json	08-06-22 15:08	Fichier JSON	85 Ko
COVID19BE_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

Les bouts de code en Python nécessite parfois un module complémentaire pour fonctionner (bibliothèques). Le petit programme en .bat disponible à la racine du projet lance des instructions 'pip' afin des les installer automatiquement (ou met à jour si nécessaire).

2. La méthode des écarts-type et des oscillateurs

Une fois le fichier 'COVID\_5BXL.json' créé il suffit de le copier dans le dossier '4.0.1 Ecart type Jordan (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'écart-type le plus proche de 0)

Ecart Type :

Jour 0+  
A  
B  
C  
D  
E

Jour 1  
A'  
B'  
C'  
D'  
E'

$$\sqrt{\frac{(A - A')^2 + (B - B')^2 + (C - C')^2 + (D - D')^2 + (E - E')^2}{5}} = \text{Ecart-type}$$

Calcul du jour 0+ :

Jour 0

Matrice probabiliste de test

Jour 0+

$$\begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} * \begin{bmatrix} [A, B, \Gamma, \Delta, E], \\ [Z, H, \Theta, I, K], \\ [\Lambda, M, N, \Xi, O], \\ [\Pi, P, \Sigma, T, Y], \\ [\Phi, X, \Psi, \Omega, \text{I}] \end{bmatrix} = \begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix}$$

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.

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

Les oscillateurs :

En partant des matrices probabilistes de base choisie à l'étape précédente il faut désormais générer une série d'autres 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 par ligne, 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 termes est égale à 0 car les oscillateurs opère par addition/soustraction, de ce fait la somme des termes des matrices probabilistes 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.

1ere passe

$$[0.0, 0.5, 0.0, 0.5, 0.0] \xrightarrow{\text{Osc, divise=10}} [-1, -1, 0, 1, 1] = [0.1, 0.4, 0.1, 0.4, 0.0]$$

2eme passe

$$[0.1, 0.4, 0.1, 0.4, 0.0] \xrightarrow{\text{Osc, divise=100}} [1, -1, 0, -1, 1] = [0.09, 0.41, 0.1, 0.39, 0.01]$$

etc...

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 laquelle de ces matrices est la meilleure base pour repartir sur une passe suivante... jusqu'à 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...)

Comme les oscillateurs génèrent un grand nombre de matrices, il a été décidé de splitter les calculs en 4. Ainsi dans le dossier il y a P1, P2, P3 et P4.  
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'

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 chaîne 'r' contient le résultat de l'écart-type et la chaîne 'mtx' la matrice associée...

Stéphane.