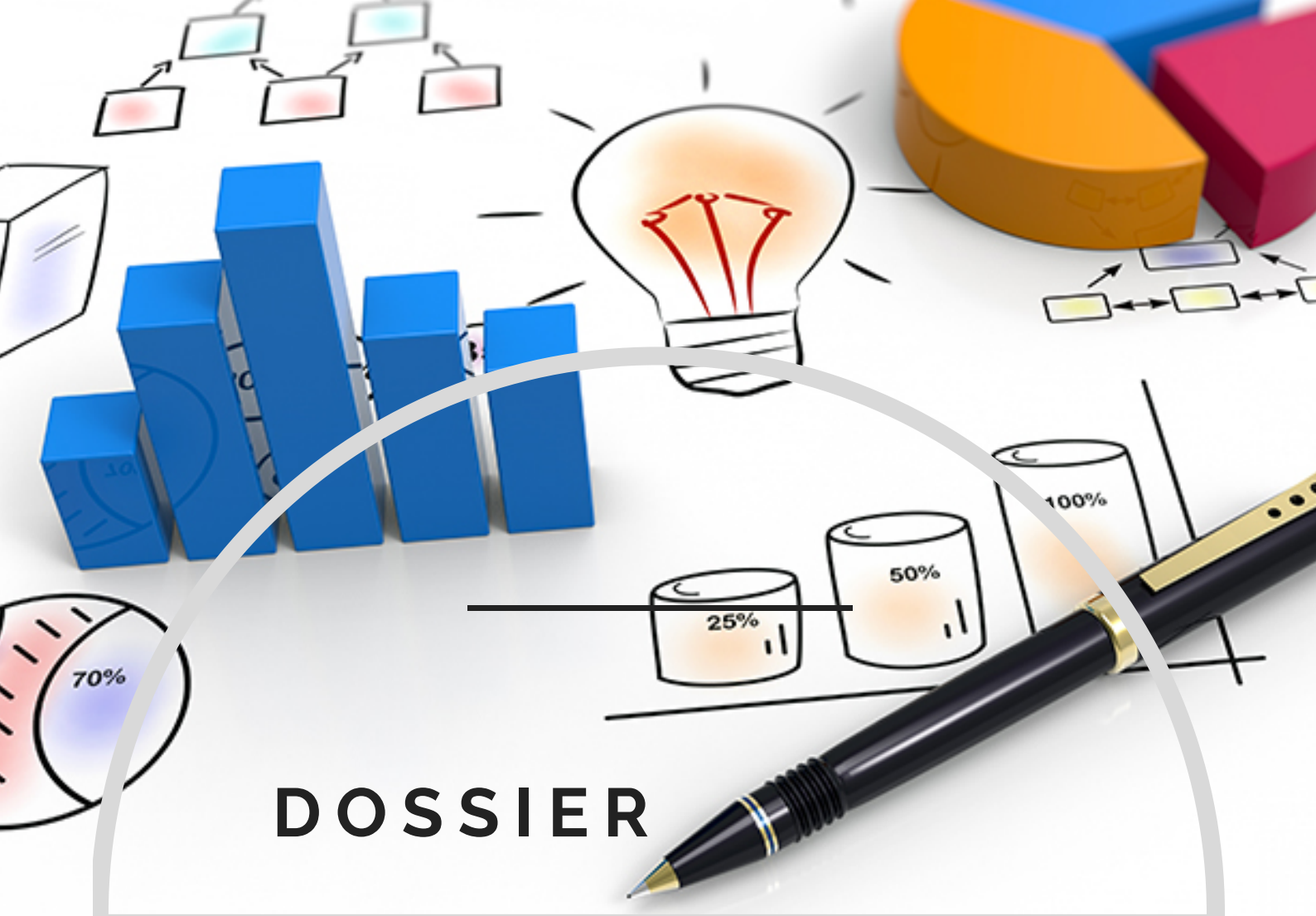


DOSSIER

ÉCONOMÉTRIE DES DONNÉES DE PANEL

M2 IDEE, UNIVERSITÉ D'ANGERS

PEDROT EMMA, SEZESTRE EMILIEN, LAMON OCÉANE

ENSEIGNANT : M. PHILLIPE COMPAIRE

Table des matières

I. Introduction

Dans le cadre de la gestion d'une crise telle que l'épidémie de COVID-19, de nombreux facteurs structurels et facteurs conjoncturels sont susceptibles d'entrer en jeu dans les capacités d'un pays à contrôler et endiguer la prolifération de la maladie. En effet, au-delà de l'intervention de l'État en termes de stratégies d'anticipation de crise et de décisions liées à l'évolution de la situation sanitaire, chaque pays est doté de diverses caractéristiques inhérentes à sa population qui, dans leur structure ou leurs comportements, font état d'une plus ou moins forte adhérence aux incitations du gouvernement, et donc d'une plus ou moins grande réussite de ces stratégies. Nous choisissons plus spécifiquement de nous focaliser sur les décès imputés au virus dans différents pays d'Europe, et tenter de voir quelles caractéristiques structurelles ou conjoncturelles exercent une influence significative sur ces derniers.

Les données de panel sont relatives à des mêmes individus et leurs caractéristiques suivis de façon régulière dans le temps. De ce fait, ce type de données possède à la fois une dimension individuelle (transversale) et temporelle (longitudinale). Dans le cadre de cette étude, nous avons choisi de nous intéresser à des caractéristiques relatives à l'épidémie de COVID-19 observées sur une base journalière pour vingt-six pays, dont la variable endogène retenue est le nombre de morts liés au virus pour 1 000 000 de personnes. Notre but est de déterminer si, les caractéristiques intrinsèques au pays et leur évolution dans le temps, telles que la composition de sa population et leur sensibilité aux gestes barrières, sont en mesure d'expliquer le nombre de morts du COVID à un instant t.

De ce fait, après avoir présenté le jeu de données qui constitue le cœur de notre analyse, nous utiliserons dans un premier temps le logiciel Stata pour réaliser notre modèle sur données longitudinales, puis nous recourrons à une méthode de machine learning en langage Python pour l'estimation de notre variable expliquée.

II. Présentation des données

Les données utilisées dans le cadre de cette étude ont été extraites d'un jeu de données rendu disponible sur Github [1] qui recense des données relatives au COVID-19 à travers cas confirmés, décès, hospitalisations et tests, ainsi que d'autres variables d'intérêt potentiel provenant de diverses sources (la principale étant *Our World in Data*).

Les données recueillies comprennent un total de 18201 observations, soit 59 variables mesurées de façon journalière du 01/06/2020 au 01/05/2022 pour 26 pays d'Europe. Les deux tableaux ci-dessous recensent respectivement les sources depuis lesquelles ont été extraites les mesures (variables regroupées par thèmes) de la base, ainsi que les pays concernés par notre analyse. La liste complète des variables utilisées est rendue disponible en annexe.

Table 1 : Origine des différentes variables, par thème

Mesures	Source	Mises à jour	Pays
Vaccinations	Our World in Data	Journalière	218
Tests & positivité	Our World in Data	N'est plus mis à jour (#2667)	193
Hôpitaux & soins intensifs	Our World in Data	Journalière	47
Cas confirmés	JHU CSSE COVID-19 Data	Journalière	219
Morts confirmés	JHU CSSE COVID-19 Data	Journalière	219
Taux de reproduction	Arroyo-Marioli F, Bullano F, Kucinskas S, Rondón-Moreno C	Journalière	196
Réponses politiques	Oxford COVID-19 Government Response Tracker	Journalière	187
Autres variables d'intérêt	Organisations internationales (UN, World Bank, OECD, IHME…)	Fixées	241

Table 2 : Pays étudiés, par ordre alphabétique

Continent	Pays
Europe	Belgique, Bulgarie, Croatie, Chypre, Danemark, Espagne, Estonie, Finlande, France, Hongrie, Irlande, Italie, Lettonie, Luxembourg, Malte, Norvège, Pays-Bas, Pologne, Portugal, Royaume-Uni, Serbie, Slovaquie, Slovénie, Suède, Suisse, Tchéquie.

III. Modèle sur données de panel

1. Estimations

Pour réaliser des estimations sur des données de panel, il faut suivre 4 étapes. Tout d'abord, il faut commencer par réaliser une analyse statistique des variables et de la stationnarité, qui se fait via l'utilisation de 3 tests de spécification, organisés en 3 étapes résumé ici. Le premier test est un test d'homogénéité. Le second test permet de retenir si un modèle de panel est pertinent, c'est-à-dire que si cette hypothèse est refusée, il faudra réaliser des estimations sur l'ensemble des individus, un à un. Pour finir, le troisième test permet de déterminer l'existence ou non d'effet individuel. Ainsi, si l'une de ces hypothèses n'est pas respectée, les résultats que l'on trouve ne seront pas pertinents.

La seconde étape consiste à tester notre modèle pour détecter s'il existe ou non, une multi colinéarité entre les variables, pour cela nous utiliserons le test VIF.

Lors de la troisième étape, nous allons estimer 9 types de modèles : (OLS, within avec effet temporel et individuel, within avec effet temporel, etc…)

Pour finir, nous réalisons des tests pour savoir quels types de modèles nous retenons, ils sont au nombre de 4 et permettent d'estimer quel modèle est le plus pertinent.

Première étape : Les tests de spécification

Premier test : Test d'indépendance/corrélation entre les individus

	__e17	__e18	__e19	__e20	__e21	__e22	__e23	__e24	__e25	__e26
__e17	1.0000									
__e18	0.0647	1.0000								
__e19	-0.2071	0.0512	1.0000							
__e20	0.0598	0.1803	-0.0512	1.0000						
__e21	-0.1401	0.2185	0.3034	-0.1083	1.0000					
__e22	-0.0069	-0.0833	0.3051	0.0673	-0.2081	1.0000				
__e23	0.0029	0.0133	0.2627	0.0248	0.1437	0.1319	1.0000			
__e24	0.1795	0.3315	0.2660	0.1878	0.1343	0.1435	0.2272	1.0000		
__e25	0.2715	-0.0873	0.1594	-0.0555	-0.0916	0.5640	0.0720	0.1185	1.0000	
__e26	-0.2409	-0.0376	0.7914	-0.2533	0.2526	0.3700	0.1485	0.1721	0.2342	1.0000

```
Breusch-Pagan LM test of independence: chi2(325) = 10640.482, Pr = 0.0000
Based on 700 complete observations over panel units
```

Comme la P-value est inférieur à 5% nous pouvons dire qu'il y a une dépendance en coupe transversale, ainsi, les résidus sont corrélés avec les individus.

Second test : Le test Pasaran : test d'indépendance/corrélation entre les individus

```
Pesaran's test of cross sectional independence =    52.659, Pr = 0.0000
```

Comme la p-value est inférieur à 5 %, nous avons la même conclusion que pour le test précédent.

Troisième test : Test d'hétéroscédasticité de Wald

```
Modified Wald test for groupwise heteroskedasticity
in fixed effect regression model

H0: sigma(i)^2 = sigma^2 for all i

chi2 (26)  =    4936.30
Prob>chi2 =      0.0000
```

Comme la p-value est inférieure à 5 %, nous pouvons dire qu'il y a de l'hétéroscédasticité (les variances des résidus sont différentes)

Seconde étape : Test sur la multi-colinéarité

Nous devons commencer par réaliser le test vif, pour vérifier qu'il n'y a pas de colinéarité dans notre modèle. Après avoir testé de nombreux modèles, celui que nous avons décidé de conserver est le suivant car tous les scores VIF sont inférieur à 10.

Variable	VIF	1/VIF
total_boos~s	4.73	0.211621
total_tests	4.60	0.217340
human_deve~x	4.14	0.241592
gdp_per_ca~a	2.81	0.355776
hospital_b~d	2.44	0.409170
population	2.34	0.427899
female_smo~s	2.16	0.463764
diabetes_p~e	2.13	0.470525
new_cases	1.81	0.552913
median_age	1.54	0.648662
positive_r~e	1.38	0.726567
stringency~x	1.37	0.731184
population~y	1.17	0.851452
tests_per_~e	1.16	0.865168
reproducti~e	1.06	0.944217
Mean VIF	2.32	

Nous avons également réalisé une table des corrélations, disponible ci-dessous pour étudier s'il n'y avait pas de corrélation trop forte entre nos variables.

	new_de„	popula~n	positi~e	string~x	hospit~d	reprod~e	total~ts	popula~y	gdp_pe~a	tests_~e	total~rs	human_~x	new_ca~s	diabet~e	median~e	female_smo~s
new_deaths„	1															
population	-0,0129	1														
positive_r~e	0,3235	-0,0731	1													
stringency~x	0,2616	0,1992	-0,2406	1												
hospital_b~d	0,2837	-0,123	0,1088	-0,1181	1											
reproducti~e	-0,1728	-0,0117	-0,0615	-0,0867	-0,0124	1										
total_tests	-0,0323	0,6202	0,0104	-0,1363	-0,1499	-0,0567	1									
population~y	-0,0728	-0,0345	-0,0787	0,0568	-0,0458	-0,0348	0,0004	1								
gdp_per_ca~a	-0,2169	-0,091	-0,0521	-0,0025	-0,4289	0,0339	-0,0286	0,0784	1							
tests_per_~e	-0,2028	-0,0797	-0,3141	0,0402	-0,0424	-0,063	-0,0535	0,0182	0,0121	1						
total_boos~s	-0,044	0,534	0,1555	-0,205	-0,153	-0,0682	0,8429	0,0236	-0,0439	-0,0899	1					
human_deve~	-0,2659	0,081	-0,0388	0,0285	-0,6449	0,0236	0,0979	0,0954	0,7305	0,0076	0,0626	1				
new_cases	0,0942	0,4067	0,2312	-0,0356	-0,0513	0,0426	0,5344	-0,0036	-0,016	-0,1098	0,6226	0,0541	1			
diabetes_p~e	0,049	-0,2475	-0,0256	0,0464	0,1202	-0,0138	-0,179	0,2073	-0,487	0,0728	-0,1344	-0,5189	-0,1101	1		
median_age	0,1421	0,2484	0,08	0,0525	0,131	-0,0058	0,0234	0,0025	-0,4474	-0,1374	0,1397	-0,2933	0,0813	0,1039	1	
female_smo~	0,1921	0,0366	0,1756	-0,098	0,6449	-0,0057	-0,0552	-0,134	-0,4422	-0,1016	-0,0508	-0,5919	0,0321	0,0929	0,1037	1

Nous pouvons constater que la plus forte des corrélations est de 0,84 entre total_tests et total_boosters, ce qui est très élevé, cependant nous décidons tout de même de garder ces 2 variables dans notre modèle, car elles n'ont a priori rien à voir. Il est également intéressant de remarquer que la seconde plus forte corrélation est entre le PIB par habitant et l'IDH, ce qui est « logique ».

Troisième étape : Estimation des 9 modèles

Nous allons désormais passer à la troisième partie qui consiste en l'estimation de 9 modèles, et en la sélection de ces derniers. Tous les résultats de ces régressions seront disponibles en annexe. Nous avons détaillé ci-dessous, le fonctionnement de 3 estimateurs.

OLS : Pour commencer notre étude nous allons réaliser une estimation par les moindres carrées ordinaires. Cette méthode est importante pour réaliser des comparaisons avec les autres estimations. En effet, les autres estimateurs que nous allons utiliser s'interprètent comme des moindres carrées ordinaires mais appliqués à une transformation des données de départ (estimateur within, between, etc…). Pour réaliser ces comparaisons, nous devons retirer le « terme constant » du modèle, nous allons centrer les données (retirer l'effet temps et l'effet individuel).

Estimateur Within: L'estimateur intra-individuelle ou « within » permet d'étudier les écarts aux moyennes individuelles, c'est-à-dire leurs variations de comportement dans le temps. Ainsi avec ce modèle, on retire l'effet individuel résiduel.

Estimateur between : L'estimateur inter-individuelle ou « between » se focalise sur les différences entre les individus. Pour ce faire, la méthode consiste à calculer les moyennes (ici du nombre de morts par jours) pour chaque individu (ici les pays sélectionnées) et d'effectuer une régression par moindres carrés ordinaires. Ainsi avec ce modèle, on retire l'effet temporel résiduel.

La régression par OLS nous donne le résultat suivant :

Source	SS	df	MS		
				Number of obs	= 18,200
				F(15, 18184)	= 720.79
Model	158880.763	15	10592.0509	Prob > F	= 0.0000
Residual	267213.638	18,184	14.6949867	R-squared	= 0.3729
				Adj R-squared	= 0.3724
Total	426094.401	18,199	23.4130667	Root MSE	= 3.8334

new_deaths_per_million	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
population	-3.58e-08	2.10e-09	-17.06	0.000	-3.99e-08	-3.17e-08
positive_rate	14.02982	.2763321	50.77	0.000	13.48819	14.57146
stringency_index	.1131215	.001996	56.68	0.000	.1092092	.1170338
hospital_beds_per_thousand	.64998	.0296332	21.93	0.000	.591896	.7080639
reproduction_rate	-2.098343	.1028442	-20.40	0.000	-2.299927	-1.896758
total_tests	2.03e-08	1.06e-09	19.08	0.000	1.82e-08	2.24e-08
population_density	-.0005328	.0001113	-4.79	0.000	-.0007509	-.0003146
gdp_per_capita	-.0000149	2.83e-06	-5.27	0.000	-.0000205	-9.37e-06
tests_per_case	-.0034304	.0001978	-17.34	0.000	-.0038181	-.0030427
total_boosters	-1.60e-07	1.05e-08	-15.15	0.000	-1.80e-07	-1.39e-07
human_development_index	-22.96263	1.404757	-16.35	0.000	-25.71608	-20.20917
new_cases	.0000174	1.66e-06	10.46	0.000	.0000141	.0000206
diabetes_prevalence	-.2829445	.0230236	-12.29	0.000	-.3280729	-.237816
median_age	.0969443	.015524	6.24	0.000	.0665157	.1273729
female_smokers	-.0979802	.008124	-12.06	0.000	-.1139041	-.0820563
_cons	17.37236	1.672036	10.39	0.000	14.09501	20.64971

Ainsi, l'ensemble de nos variables sont significatives, nous obtenons un R^2 de 0,3729, ce qui signifie que l'on explique 37,29 % de la variance avec notre modèle.

Pour comparer nos modèles, nous avons réalisé le tableau récapitulatif ci-dessous :

MODELE	OLS	Within2	WithinT	WithinP	Aleatoire2	AléatoireP	AléatoireT	BetweenP	BetweenT
N	18200	18200	18200	18200	18200	18200	18200	18200	18200
R^2	0,3729	0,6492							
R^2 ajusté	0,3724	0,6345							
rho			0,2197	0,21144	0,0368				
Corr			-0,0517	-0,1247	0				
Sd(residual)						3,831719	3,592		
Within								0,0025	0,025
Between								0,9324	0,9324
Overall								0,0052	0,052

2. Tests

Ensuite nous devrons réaliser un test de Fischer, qui nous permettra de déterminer si les OLS donnent ou non un résultat significatif. Si H1 est accepté, alors il faudra alors cela voudra dire qu'il y a soit un effet individuel, soit un effet temporel, non mesurable avec une régression par OLS classique.

Pour finir, nous devrons réaliser le test d'Hausman, qui permet de tester s' il existe une présence éventuelle d'une corrélation entre les effets individuels et les variables explicatives. Si l'on accepte H0, le modèle devra être spécifié avec des effets individuels aléatoires, nous estimons alors par les GLS. Si l'on accepte H1, on spécifie avec des effets individuels fixes, par l'estimateur « within ».

Premier test : test de Breusch-Pagan (OLS vs effets aléatoires individuels)

```
Breusch and Pagan Lagrangian multiplier test for random effects

        new_deaths_per_million[code,t] = Xb + u[code] + e[code,t]

        Estimated results:
                         |       Var     SD = sqrt(Var)
                ---------+-----------------------------
                new_dea.. |   23.41307       4.838705
                        e |   13.92536       3.731669
                        u |   .5321377       .7294777

        Test: Var(u) = 0
                                  chibar2(01) = 14564.26
                               Prob > chibar2 =   0.0000
```

La probabilité de la statistique de test est inférieure à 0.05, alors l'on choisit le modèle à effets aléatoires.

Premier test (bis) : test de Breusch-Pagan (OLS vs effets aléatoires fixes)

```
Estimated results:
                |        Var     SD = sqrt(Var)
----------------+------------------------------
       new_dea.. |   23.41307       4.838705
              e |   12.88776       3.589952
              u |   .7218445       .8496143

Test: Var(u) = 0
                          chibar2(01) =  2354.06
                      Prob > chibar2 =   0.0000
```

De la même façon, la probabilité de la statistique de test étant inférieure à 0.05, l'on choisit une fois de plus le modèle à effets aléatoires plutôt que les OLS.

Second test : test d'Hausman (effet aléatoire contre effet fixe)

```
Test of H0: Difference in coefficients not systematic

    chi2(7) = (b-B)'[(V_b-V_B)^(-1)](b-B)
            = 736.28
Prob > chi2 = 0.0000
```

Après avoir éliminé les OLS, il est nécessaire de choisir entre un modèle à effets aléatoires (H0) et un modèle à effets fixes (H1). C'est le principe du test d'Hausman. On a prob>chi2 < 5%, donc on rejette l'hypothèse nulle et on choisit donc un modèle à effets fixes.

Troisième test : test d'Hausman (effet fixe contre between)

```
          B = Inconsistent under Ha, efficient under H0; obtained from xtreg.

Test of H0: Difference in coefficients not systematic

    chi2(4) = (b-B)'[(V_b-V_B)^(-1)](b-B)
            = 326.00
Prob > chi2 = 0.0000
```

Quatrième test : test pour effets temporels présents ou non dans le modèle within

```
-----------------+------------------------------------------------------------
         sigma_u |  1.6860394
         sigma_e |  3.5899521
             rho |  .18071463   (fraction of variance due to u_i)
------------------------------------------------------------------------------
F test that all u_i=0: F(699, 17485) = 4.65                 Prob > F = 0.0000
```

On a Prob>F < .05 donc aucun effet fixe temporel n'est nécessaire dans le modèle.

IV. Prédiction à l'aide d'un algorithme de machine learning

Cette seconde partie aura vocation à prédire notre variable expliquée, soit le nombre total de morts pour un million de personnes, à l'aide d'un algorithme de machine learning. Pour ce faire, nous avons utilisé le package *sklearn* sous Python. Le script utilisé est rendu disponible en annexe.

L'algorithme utilisé a été construit de sorte à répartir les différentes observations en deux groupes : celles à prédire (groupe test), et celles sur lequel l'algorithme va s'entraîner pour prédire leur valeur au mieux, compte tenu de celles des variables explicatives. Nous avons choisi de composer nos deux groupes train et test à hauteur respective de 80% (14500 observatiobs) et 20% (3640 observations) de notre échantillon total, les observations étant réparties de façon purement aléatoire (une variable ID a été créé de sorte à affecter un identifiant unique pour le croisement d'une date et d'un pays en particulier).

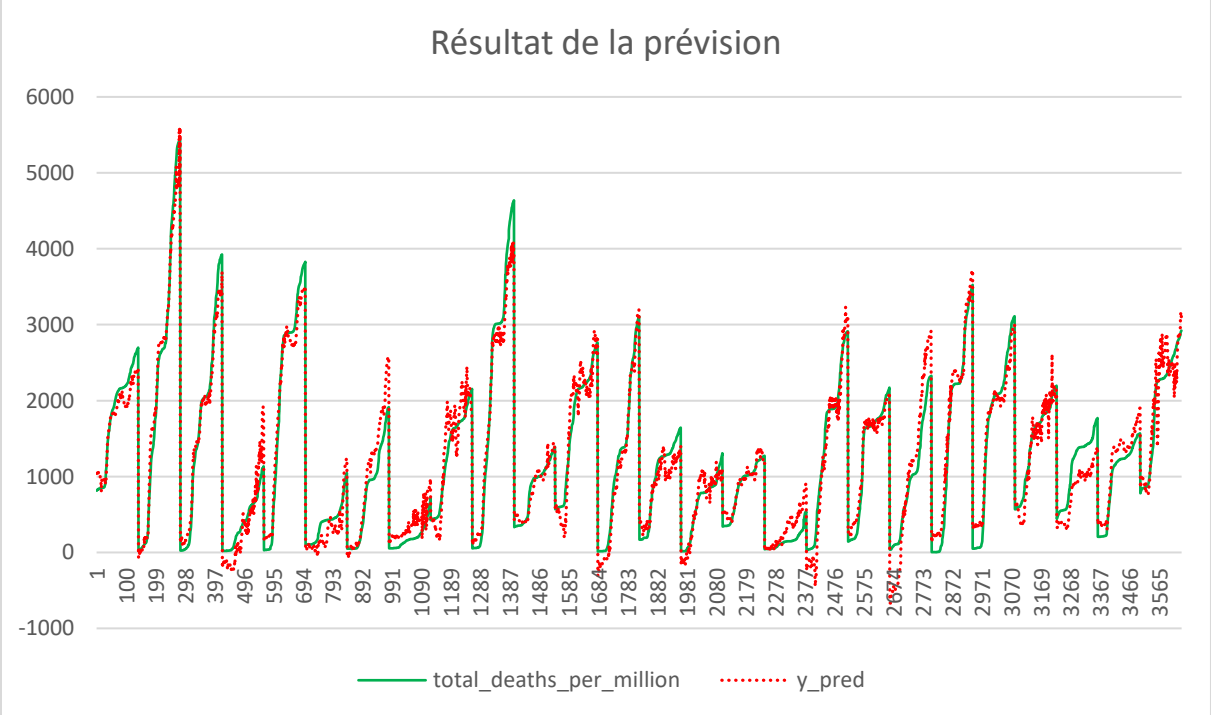
Les variables ont également été réparties dans deux groupes distincts, le groupe Y contenant la variable à prédire et le groupe X contenant toutes les variables utilisées pour comprendre la variation de Y. Les variables peuvent alors être affectées dans quatre catégories :

- x_train : utilisées par l'algorithme dans sa phase d'entraînement, pour comprendre le fonctionnement et l'impact des variables
- x_test : utilisées pour tester le fonctionnement de l'algorithme à l'issue de son entraînement
- y_train : valeurs empiriques utilisées pour l'entraînement de l'algorithme
- y_test : permet de valider ou non les résultats
- y_pred : résultats de la prévision

A l'issue de la mise en fonctionnement de l'algorithme, nous obtenons des valeurs prédites particulièrement fiables avec entre autres un R^2 d'une valeur de 0.954. Les résultats obtenus peuvent être appréciés par le biais du graphique ci-dessous. Nous constatons aisément que le tracé de la courbe des valeurs prédites (ligne pointillée rouge) est très fidèle à celui des valeurs observées (ligne pleine verte). Les prédictions des quarante premières observations sont également répertoriées dans le tableau ci-dessous.

ID	total_deaths_per_million	y_pred
1	813,835	1038,98
2	821,9	1049,19
3	824,13	1043,23
4	826,704	1045,03
5	830,736	1054,01
6	831,852	1019,42
7	837,428	1059,71
8	838,286	1062,53
9	839,23	1042,96
10	840,774	1013,94
11	841,461	1020,08
12	842,576	1016,35
13	844,635	941,669
14	845,836	917,421
15	846,951	907,204
16	849,354	797,78
17	853,128	938,442
18	856,903	935,745
19	849,182	974,13
20	850,126	983,432

ID	total_deaths_per_million	y_pred
21	851,327	983,512
22	851,67	952,571
23	852,356	955,467
24	853,472	947,794
25	855,273	966,465
26	859,306	980,557
27	865,826	970,628
28	888,733	955,326
29	893,366	950,892
30	921,163	898,924
31	1040,329	925,732
32	1074,132	957,004
33	1090,261	970,989
34	1133,844	1105,22
35	1227,101	1164,76
36	1303,715	1299,48
37	1412,243	1421,74
38	1461,317	1509,35
39	1501,983	1546,4
40	1540,075	1555,17

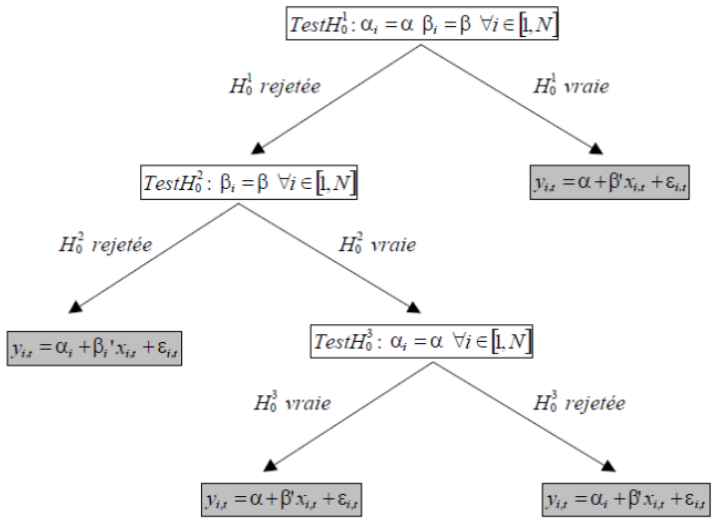


V. Annexes

[1] Source des données : « Covid-19-Data/Public/Data at Master · Owid/Covid-19-Data ». *GitHub*, https://github.com/owid/covid-19-data.

[2] Liste des variables : « Data on COVID-19 (coronavirus) by Our World in Data ». *Github*, https://github.com/owid/covid-19-data/tree/master/public/data.

[3] Test de spécification



[4] Estimation 1 : Modèle OLS

Source	SS	df	MS	Number of obs	=	18,200
				F(15, 18184)	=	720.79
Model	158880.763	15	10592.0509	Prob > F	=	0.0000
Residual	267213.638	18,184	14.6949867	R-squared	=	0.3729
				Adj R-squared	=	0.3724
Total	426094.401	18,199	23.4130667	Root MSE	=	3.8334

new_deaths_per_million	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
population	-3.58e-08	2.10e-09	-17.06	0.000	-3.99e-08	-3.17e-08
positive_rate	14.02982	.2763321	50.77	0.000	13.48819	14.57146
stringency_index	.1131215	.001996	56.68	0.000	.1092092	.1170338
hospital_beds_per_thousand	.64998	.0296332	21.93	0.000	.591896	.7080639
reproduction_rate	-2.098343	.1028442	-20.40	0.000	-2.299927	-1.896758
total_tests	2.03e-08	1.06e-09	19.08	0.000	1.82e-08	2.24e-08
population_density	-.0005328	.0001113	-4.79	0.000	-.0007509	-.0003146
gdp_per_capita	-.0000149	2.83e-06	-5.27	0.000	-.0000205	-9.37e-06
tests_per_case	-.0034304	.0001978	-17.34	0.000	-.0038181	-.0030427
total_boosters	-1.60e-07	1.05e-08	-15.15	0.000	-1.80e-07	-1.39e-07
human_development_index	-22.96263	1.404757	-16.35	0.000	-25.71608	-20.20917
new_cases	.0000174	1.66e-06	10.46	0.000	.0000141	.0000206
diabetes_prevalence	-.2829445	.0230236	-12.29	0.000	-.3280729	-.237816
median_age	.0969443	.015524	6.24	0.000	.0665157	.1273729
female_smokers	-.0979802	.008124	-12.06	0.000	-.1139041	-.0820563
_cons	17.37236	1.672036	10.39	0.000	14.09501	20.64971

[5] Estimation 2 : Modèle whitin avec effet temporel et individuel

Source	SS	df	MS	Number of obs	=	18,200
				F(732, 17468)	=	44.16
Model	391575.964	732	534.939842	Prob > F	=	0.0000
Residual	211622.175	17,468	12.1148486	R-squared	=	0.6492
				Adj R-squared	=	0.6345
Total	603198.139	18,200	33.1427549	Root MSE	=	3.4806

new_deaths_per_million	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
population	1.86e-08	4.48e-09	4.16	0.000	9.87e-09	2.74e-08
positive_rate	12.16973	.3589713	33.90	0.000	11.46611	12.87335
stringency_index	.0801044	.0031805	25.19	0.000	.0738703	.0863385
hospital_beds_per_thousand	.2904679	.0600837	4.83	0.000	.1726979	.408238
reproduction_rate	-2.764855	.1161675	-23.80	0.000	-2.992555	-2.537155
total_tests	-2.34e-10	1.19e-09	-0.20	0.844	-2.56e-09	2.09e-09
population_density	-.0001852	.0001226	-1.51	0.131	-.0004255	.000055
gdp_per_capita	7.57e-06	4.02e-06	1.88	0.060	-3.09e-07	.0000155
tests_per_case	-.0016632	.0002052	-8.11	0.000	-.0020653	-.001261
total_boosters	-1.63e-08	1.11e-08	-1.47	0.142	-3.81e-08	5.48e-09
human_development_index	0	(omitted)				
new_cases	.0000146	1.64e-06	8.87	0.000	.0000113	.0000178
diabetes_prevalence	.1382279	.0494714	2.79	0.005	.0412591	.2351967
median_age	-.0959958	.0205884	-4.66	0.000	-.1363512	-.0556404
female_smokers	-.0396052	.0111362	-3.56	0.000	-.0614332	-.0177772
_Idate_22068	.4886722	.9653796	0.51	0.613	-1.403568	2.380913
_Idate_22069	.5105854	.9653856	0.53	0.597	-1.381667	2.402838
_Idate_22070	.3732249	.9654145	0.39	0.699	-1.519084	2.265534
_Idate_22071	.8090639	.9655426	0.84	0.402	-1.083496	2.701624
_Idate_22072	.6345835	.9655497	0.66	0.511	-1.25799	2.527157
_Idate_22073	.5624707	.9655609	0.58	0.560	-1.330125	2.455066
_Idate_22074	.9377115	.9656388	0.97	0.332	-.9550369	2.83046
_Idate_22075	.9724484	.9656739	1.01	0.314	-.9203689	2.865266

[6] Estimation 3 : Modèle whitin avec effet temporel

```
Fixed-effects (within) regression               Number of obs      =     18,200
Group variable: code                            Number of groups   =         26

R-squared:                                      Obs per group:
     Within  = 0.4264                                         min =        700
     Between = 0.0155                                         avg =      700.0
     Overall = 0.3636                                         max =        700

                                                F(706,17468)       =      18.39
corr(u_i, Xb) = -0.0517                         Prob > F           =     0.0000
```

new_deaths_per_million	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
population	0	(omitted)				
positive_rate	12.16973	.3589713	33.90	0.000	11.46611	12.87335
stringency_index	.0801044	.0031805	25.19	0.000	.0738703	.0863385
hospital_beds_per_thousand	0	(omitted)				
reproduction_rate	-2.764855	.1161675	-23.80	0.000	-2.992555	-2.537155
total_tests	-2.34e-10	1.19e-09	-0.20	0.844	-2.56e-09	2.09e-09
population_density	0	(omitted)				
gdp_per_capita	0	(omitted)				
tests_per_case	-.0016632	.0002052	-8.11	0.000	-.0020653	-.001261
total_boosters	-1.63e-08	1.11e-08	-1.47	0.142	-3.81e-08	5.48e-09
human_development_index	0	(omitted)				
new_cases	.0000146	1.64e-06	8.87	0.000	.0000113	.0000178
diabetes_prevalence	0	(omitted)				
median_age	0	(omitted)				
female_smokers	0	(omitted)				
date						
22068	.4886722	.9653796	0.51	0.613	-1.403568	2.380913

22761	2.13976	.9801991	2.18	0.029	.218472	4.061048
22762	2.557663	.9803713	2.61	0.009	.6360373	4.479288
22763	4.469808	.9800052	4.56	0.000	2.5489	6.390716
22764	1.630217	.9802801	1.66	0.096	-.2912295	3.551664
22765	1.137156	.9800639	1.16	0.246	-.7838676	3.058179
22766	1.015619	.9803162	1.04	0.300	-.9058989	2.937136
_cons	-1.789114	.7221495	-2.48	0.013	-3.204599	-.373629
sigma_u	1.8469109					
sigma_e	3.4806391					
rho	.21970215	(fraction of variance due to u_i)				

F test that all u_i=0: F(25, 17468) = 188.15 Prob > F = 0.0000

[7] Estimation 4 : Modèle within avec effet individuel

Fixed-effects (within) regression Number of obs = 18,200
Group variable: code Number of groups = 26

R-squared: Obs per group:
Within = 0.3143 min = 700
Between = 0.0134 avg = 700.0
Overall = 0.2554 max = 700

F(7,18167) = 1189.70
corr(u_i, Xb) = -0.1247 Prob > F = 0.0000

new_deaths_per_million	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
population	0	(omitted)				
positive_rate	14.69447	.2766591	53.11	0.000	14.15219	15.23675
stringency_index	.1131199	.0020475	55.25	0.000	.1091066	.1171332
hospital_beds_per_thousand	0	(omitted)				
reproduction_rate	-2.256588	.1009093	-22.36	0.000	-2.45438	-2.058796
total_tests	1.21e-08	1.20e-09	10.06	0.000	9.73e-09	1.44e-08
population_density	0	(omitted)				
gdp_per_capita	0	(omitted)				
tests_per_case	-.0040231	.0001982	-20.30	0.000	-.0044116	-.0036346
total_boosters	-1.25e-07	1.11e-08	-11.24	0.000	-1.46e-07	-1.03e-07
human_development_index	0	(omitted)				
new_cases	.0000195	1.67e-06	11.71	0.000	.0000162	.0000228
diabetes_prevalence	0	(omitted)				
median_age	0	(omitted)				
female_smokers	0	(omitted)				
_cons	-1.302042	.1761867	-7.39	0.000	-1.647385	-.9566996
sigma_u	1.9322264					
sigma_e	3.7314638					
rho	.21144198	(fraction of variance due to u_i)				

F test that all u_i=0: F(25, 18167) = 175.25 Prob > F = 0.0000

[8] Estimation 5 : Modèle effets aléatoires produit

```
Random-effects GLS regression                   Number of obs     =     18,200
Group variable: code                            Number of groups  =         26

R-squared:                                      Obs per group:
     Within  = 0.3143                                         min =        700
     Between = 0.7378                                         avg =      700.0
     Overall = 0.3702                                         max =        700

                                                Wald chi2(15)     =    8424.51
corr(u_i, X) = 0 (assumed)                      Prob > chi2       =     0.0000
```

new_deaths_per_million	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
population	-2.75e-08	8.07e-09	-3.41	0.001	-4.33e-08	-1.17e-08
positive_rate	14.66748	.2765413	53.04	0.000	14.12547	15.20949
stringency_index	.1131302	.0020447	55.33	0.000	.1091226	.1171378
hospital_beds_per_thousand	.6439466	.1499256	4.30	0.000	.3500979	.9377953
reproduction_rate	-2.250704	.100944	-22.30	0.000	-2.44855	-2.052857
total_tests	1.25e-08	1.19e-09	10.47	0.000	1.02e-08	1.48e-08
population_density	-.0004943	.0005665	-0.87	0.383	-.0016046	.0006159
gdp_per_capita	-.0000178	.0000144	-1.24	0.217	-.000046	.0000104
tests_per_case	-.0039969	.0001981	-20.17	0.000	-.0043852	-.0036086
total_boosters	-1.27e-07	1.11e-08	-11.45	0.000	-1.48e-07	-1.05e-07
human_development_index	-23.77122	7.106284	-3.35	0.001	-37.69928	-9.843156
new_cases	.0000194	1.67e-06	11.67	0.000	.0000162	.0000227
diabetes_prevalence	-.2959946	.1174341	-2.52	0.012	-.5261612	-.065828
median_age	.0518704	.0754416	0.69	0.492	-.0959924	.1997332
female_smokers	-.1108807	.0406183	-2.73	0.006	-.1904912	-.0312703
_cons	20.64182	8.298106	2.49	0.013	4.377833	36.90581
sigma_u	.72947772					
sigma_e	3.7316693					
rho	.03680705	(fraction of variance due to u_i)				

[9] Estimation 6 : Modèle two-way effet aleatoire

```
Mixed-effects ML regression                     Number of obs     =     18,200
                                                Wald chi2(15)     =   10821.42
Log likelihood = -50272.986                     Prob > chi2       =     0.0000
```

new_deaths_per_million	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
population	-3.58e-08	2.10e-09	-17.07	0.000	-3.99e-08	-3.17e-08
positive_rate	14.02982	.2762106	50.79	0.000	13.48846	14.57119
stringency_index	.1131215	.0019951	56.70	0.000	.1092112	.1170318
hospital_beds_per_thousand	.64998	.0296202	21.94	0.000	.5919254	.7080345
reproduction_rate	-2.098343	.102799	-20.41	0.000	-2.299825	-1.89686
total_tests	2.03e-08	1.06e-09	19.09	0.000	1.82e-08	2.24e-08
population_density	-.0005328	.0001112	-4.79	0.000	-.0007508	-.0003147
gdp_per_capita	-.0000149	2.83e-06	-5.27	0.000	-.0000205	-9.37e-06
tests_per_case	-.0034304	.0001977	-17.35	0.000	-.0038179	-.0030428
total_boosters	-1.60e-07	1.05e-08	-15.16	0.000	-1.80e-07	-1.39e-07
human_development_index	-22.96263	1.40414	-16.35	0.000	-25.71469	-20.21056
new_cases	.0000174	1.66e-06	10.47	0.000	.0000141	.0000206
diabetes_prevalence	-.2829445	.0230135	-12.29	0.000	-.32805	-.2378389
median_age	.0969443	.0155172	6.25	0.000	.0665311	.1273575
female_smokers	-.0979802	.0081205	-12.07	0.000	-.113896	-.0820644
_cons	17.37236	1.671301	10.39	0.000	14.09667	20.64805

Random-effects parameters	Estimate	Std. err.	[95% conf.	interval]
sd(Residual)	3.831719	.0200837	3.792557	3.871285

[10] Estimation 7 : Modèle effets aléatoires temporels

```
Mixed-effects ML regression                     Number of obs     =     18,200
Group variable: date                            Number of groups  =        700
                                                Obs per group:
                                                              min =         26
                                                              avg =       26.0
                                                              max =         26
                                                Wald chi2(15)     =    7285.05
Log likelihood = -49669.148                     Prob > chi2       =     0.0000
```

new_deaths_per_million	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
population	-2.89e-08	2.02e-09	-14.29	0.000	-3.28e-08	-2.49e-08
positive_rate	11.95298	.3127613	38.22	0.000	11.33998	12.56598
stringency_index	.0944473	.0025304	37.32	0.000	.0894878	.0994069
hospital_beds_per_thousand	.6582866	.027813	23.67	0.000	.6037743	.712799
reproduction_rate	-2.388315	.1124521	-21.24	0.000	-2.608717	-2.167913
total_tests	1.28e-08	1.03e-09	12.41	0.000	1.08e-08	1.48e-08
population_density	-.0005957	.0001045	-5.70	0.000	-.0008005	-.0003909
gdp_per_capita	-.0000147	2.66e-06	-5.55	0.000	-.00002	-9.54e-06
tests_per_case	-.0017538	.0001984	-8.84	0.000	-.0021426	-.0013649
total_boosters	-8.93e-08	1.04e-08	-8.62	0.000	-1.10e-07	-6.90e-08
human_development_index	-21.32683	1.321514	-16.14	0.000	-23.91695	-18.73671
new_cases	.0000135	1.62e-06	8.30	0.000	.0000103	.0000166
diabetes_prevalence	-.2693782	.021654	-12.44	0.000	-.3118191	-.2269372
median_age	.0995836	.0146351	6.80	0.000	.0708993	.1282678
female_smokers	-.0858784	.0077071	-11.14	0.000	-.1009841	-.0707728
_cons	16.63673	1.581509	10.52	0.000	13.53703	19.73643

Random-effects parameters	Estimate	Std. err.	[95% conf.	interval]
date: Identity				
sd(_cons)	1.42503	.0499138	1.330484	1.526296
sd(Residual)	3.592462	.0192431	3.554943	3.630376

```
LR test vs. linear model: chibar2(01) = 1207.68       Prob >= chibar2 = 0.0000
```

[11] Estimation 8 : Modèle between sur les individus

```
Between regression (regression on group means)  Number of obs     =     18,200
Group variable: code                            Number of groups  =         26

R-squared:                                      Obs per group:
     Within  = 0.0025                                         min =        700
     Between = 0.9324                                         avg =      700.0
     Overall = 0.0052                                         max =        700

                                                F(15,10)          =       9.20
sd(u_i + avg(e_i.)) = .742988                   Prob > F          =     0.0006
```

new_deaths_per_million	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
population	1.24e-08	4.63e-08	0.27	0.794	-9.08e-08	1.16e-07
positive_rate	4.099495	7.375831	0.56	0.591	-12.33488	20.53387
stringency_index	.1096456	.0430726	2.55	0.029	.0136739	.2056172
hospital_beds_per_thousand	.6126222	.2101142	2.92	0.015	.1444585	1.080786
reproduction_rate	12.02084	5.464793	2.20	0.052	-.1554811	24.19716
total_tests	5.44e-08	1.86e-08	2.93	0.015	1.30e-08	9.58e-08
population_density	-.0000168	.0007837	-0.02	0.983	-.0017629	.0017294
gdp_per_capita	-.0000143	.0000159	-0.90	0.389	-.0000498	.0000211
tests_per_case	-.0016143	.0052342	-0.31	0.764	-.0132767	.0100482
total_boosters	-4.52e-07	3.75e-07	-1.21	0.255	-1.29e-06	3.83e-07
human_development_index	-16.18861	9.469941	-1.71	0.118	-37.28895	4.911736
new_cases	-.0001375	.000062	-2.22	0.051	-.0002756	6.21e-07
diabetes_prevalence	-.2755939	.1454192	-1.90	0.087	-.599608	.0484203
median_age	.2876649	.1022718	2.81	0.018	.0597891	.5155407
female_smokers	.0059088	.0556597	0.11	0.918	-.1181088	.1299264
_cons	-13.23662	11.1829	-1.18	0.264	-38.15368	11.68043

[12] Estimation 9 : Modèle between sur le temps

```
Between regression (regression on group means)  Number of obs     =     18,200
Group variable: code                            Number of groups  =         26

R-squared:                                      Obs per group:
     Within  = 0.0025                                         min =        700
     Between = 0.9324                                         avg =      700.0
     Overall = 0.0052                                         max =        700

                                                F(15,10)          =       9.20
sd(u_i + avg(e_i.)) = .742988                   Prob > F          =     0.0006
```

new_deaths_per_million	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
population	1.24e-08	4.63e-08	0.27	0.794	-9.08e-08	1.16e-07
positive_rate	4.099495	7.375831	0.56	0.591	-12.33488	20.53387
stringency_index	.1096456	.0430726	2.55	0.029	.0136739	.2056172
hospital_beds_per_thousand	.6126222	.2101142	2.92	0.015	.1444585	1.080786
reproduction_rate	12.02084	5.464793	2.20	0.052	-.1554811	24.19716
total_tests	5.44e-08	1.86e-08	2.93	0.015	1.30e-08	9.58e-08
population_density	-.0000168	.0007837	-0.02	0.983	-.0017629	.0017294
gdp_per_capita	-.0000143	.0000159	-0.90	0.389	-.0000498	.0000211
tests_per_case	-.0016143	.0052342	-0.31	0.764	-.0132767	.0100482
total_boosters	-4.52e-07	3.75e-07	-1.21	0.255	-1.29e-06	3.83e-07
human_development_index	-16.18861	9.469941	-1.71	0.118	-37.28895	4.911736
new_cases	-.0001375	.000062	-2.22	0.051	-.0002756	6.21e-07
diabetes_prevalence	-.2755939	.1454192	-1.90	0.087	-.599608	.0484203
median_age	.2876649	.1022718	2.81	0.018	.0597891	.5155407
female_smokers	.0059088	.0556597	0.11	0.918	-.1181088	.1299264
date						
22068	0	(omitted)				
22069	0	(omitted)				
22761	0	(omitted)				
22762	0	(omitted)				
22763	0	(omitted)				
22764	0	(omitted)				
22765	0	(omitted)				
22766	0	(omitted)				
_cons	-13.23662	11.1829	-1.18	0.264	-38.15368	11.68043

[13] Test d'Hausman Estimation 2 – Estimation 3

	Coefficients			
	(b) whithin2	(B) whithinT	(b-B) Difference	sqrt(diag(V_b-V_B)) Std. err.
positive_r~e	12.16973	12.16973	1.32e-11	.
stringency~x	.0801044	.0801044	-1.57e-13	.
reproducti~e	-2.764855	-2.764855	-4.29e-12	.
total_tests	-2.34e-10	-2.34e-10	-1.92e-20	1.73e-15
tests_per_~e	-.0016632	-.0016632	-1.37e-15	.
total_boos~s	-1.63e-08	-1.63e-08	-4.84e-21	9.55e-15
new_cases	.0000146	.0000146	-5.00e-18	2.23e-13

b = Consistent under H0 and Ha; obtained from **regress**.
B = Inconsistent under Ha, efficient under H0; obtained from **xtreg**.

Test of H0: Difference in coefficients not systematic

```
chi2(3) = (b-B)'[(V_b-V_B)^(-1)](b-B)
        = -0.00
```

[14] Script Python

```
# -*- coding: utf-8 -*-
"""
Created on Wed Jan 18 15:54:06 2023

@author: EMMA
"""

# ------------------------------------------------ IMPORTATION DONNEES & PACKAGES ---
------------------------------------------------ #
```

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import GroupKFold
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
import numpy as np

# importation des données
data = pd.read_excel("D:/M2 IDEE/S3/Données de Panel/plop.xlsx")

# ---------------------------------------------------- TEST VIF ------------------
---------------------------------- #

#connaitre le nom des variables du dataframe pour les utiliser dans le test VIF
data.columns

#selection des variables pour le test
X = data[['total_cases',
       'new_cases', 'new_cases_smoothed', 'total_deaths', 'new_deaths',
       'new_deaths_smoothed', 'total_cases_per_million',
       'new_cases_per_million', 'new_cases_smoothed_per_million',
       'new_deaths_per_million',
       'new_deaths_smoothed_per_million', 'reproduction_rate', 'hosp_patients',
       'hosp_patients_per_million', 'total_tests', 'new_tests',
       'total_tests_per_thousand', 'new_tests_per_thousand',
       'new_tests_smoothed', 'new_tests_smoothed_per_thousand',
       'positive_rate', 'tests_per_case', 'total_vaccinations',
       'people_vaccinated', 'people_fully_vaccinated', 'total_boosters',
       'new_vaccinations', 'new_vaccinations_smoothed',
       'total_vaccinations_per_hundred', 'people_vaccinated_per_hundred',
       'people_fully_vaccinated_per_hundred', 'total_boosters_per_hundred',
       'new_vaccinations_smoothed_per_million',
       'new_people_vaccinated_smoothed',
       'new_people_vaccinated_smoothed_per_hundred', 'stringency_index',
       'population_density', 'median_age', 'aged_65_older', 'aged_70_older',
       'gdp_per_capita', 'extreme_poverty', 'cardiovasc_death_rate',
       'diabetes_prevalence', 'female_smokers', 'male_smokers',
       'hospital_beds_per_thousand', 'life_expectancy',
       'human_development_index', 'population',
       'excess_mortality_cumulative_absolute', 'excess_mortality_cumulative',
       'excess_mortality', 'excess_mortality_cumulative_per_million']]

#création d'un dataframe pour contenir les valeurs du VIF
vif = pd.DataFrame()
vif["VIF   Factor"]   =   [variance_inflation_factor(X.values,   i)   for   i   in
range(X.shape[1])]
vif["features"] = X.columns

print(vif)

# ---------------------------------------------------- SELECTION DES VARIABLES ---
------------------------------------------------- #
#Selon les résulats du test VIF, nous allons conserver les variables ayant eu une
valeur inférieure à 11

#On compte le nb de variable ayant un VIF < 11
selection = vif["VIF Factor"]<11
selection = pd.DataFrame(selection)
selection["VIF Factor"].value_counts() #on remarque ici, que nous avons 12 variables
avec un VIF inférieur à 11

#on regroupe le dataframe des résultats vif avec celui qui nous indique les variables
a conserver
```

```
d = selection.merge(vif, how='inner', left_index=True, right_index=True)

#on récupère l'intitulé des variables a conserver
selected_rows = d[d['VIF Factor_y'] < 11]
print(selected_rows)
liste_des_variables = selected_rows["features"].to_list()
liste_des_variables

#on peut copier le résultat de 'liste_des_variables' pour intégrer les valeurs dans
l'algorithme de machine learning

# ------------------------------------------------------ MACHINE LEARNING ----------
------------------------------------------- #
re_model = LinearRegression(fit_intercept=False)

#définition du groupe de panel
group = data["id"]

#définition de la variable a prédire et des variables explicarives
X = data[['new_cases',
 'new_cases_per_million',
 'new_deaths_per_million',
 'total_tests_per_thousand',
 'positive_rate',
 'tests_per_case',
 'new_vaccinations',
 'new_vaccinations_smoothed_per_million',
 'new_people_vaccinated_smoothed',
 'new_people_vaccinated_smoothed_per_hundred',
 'population_density',
 'excess_mortality']]
y = data["total_deaths_per_million"]

# Create an instance of the GroupKFold splitter
gkf = GroupKFold(n_splits=5)

# Iterate over the splits
for train_index, test_index in gkf.split(X, y, group):
    # Get the training and testing data
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]
    group_train, group_test = group.iloc[train_index], group.iloc[test_index]

    # Fit the model on the training data
    re_model.fit(X_train, y_train)

    # Make predictions on the testing data
    y_pred = re_model.predict(X_test)
```