

Analyse de la situation de la France suite à l'épidémie de COVID-19

Clustering using K-means

Etudiants : Owen ALLEMANG & Antoine BURGLIN

Date : 12/05/2020

Le but de cette étude est d'analyser l'évolution du COVID-19, plus particulièrement d'observer la situation de la France face à d'autres pays du monde avant son confinement et après son confinement.

Pour ce faire, nous avons créé deux bases de données à partir des rapports journaliers issue du site de la WHO (World Health Organization), NYTimes et (CSSE) Johns Hopkins University. Nous avons créé des bases de données pour deux dates :

- au 15/03/2020 (avant confinement),
- au 10/05/2020 (après confinement).

Chaque base de données comporte trois attributs : le pays, le nombre de cas infectés et le nombre de morts.

Ensuite nous avons utilisé un algorithme de clustering (using K-means), en usant de la librairie *Sklearn*. Il s'agit d'apprentissage non-supervisé.

Dans cet algorithme nous utilisons $K = 3$, ce qui correspond à trois clusters :

- Situation de Faible intensité : en bleu - cluster 0,
- Situation de Moyenne intensité : en vert - cluster 1,
- Situation de Forte intensité : en rouge - cluster 2.

Avec notre algorithme de clustering et nos bases de données, nous avons entraîné l'algorithme avec tous les pays du monde sauf la France. Une fois cet entraînement terminé, nous avons demandé à l'algorithme de nous fournir un cluster d'appartenance pour la France aux deux dates différentes connaissant les chiffres du nombre de cas infectés et du nombre de mort :

Au 15/03/2020 :

- Infectés : 20 096
- Morts : 395

Au 10/05/2020 :

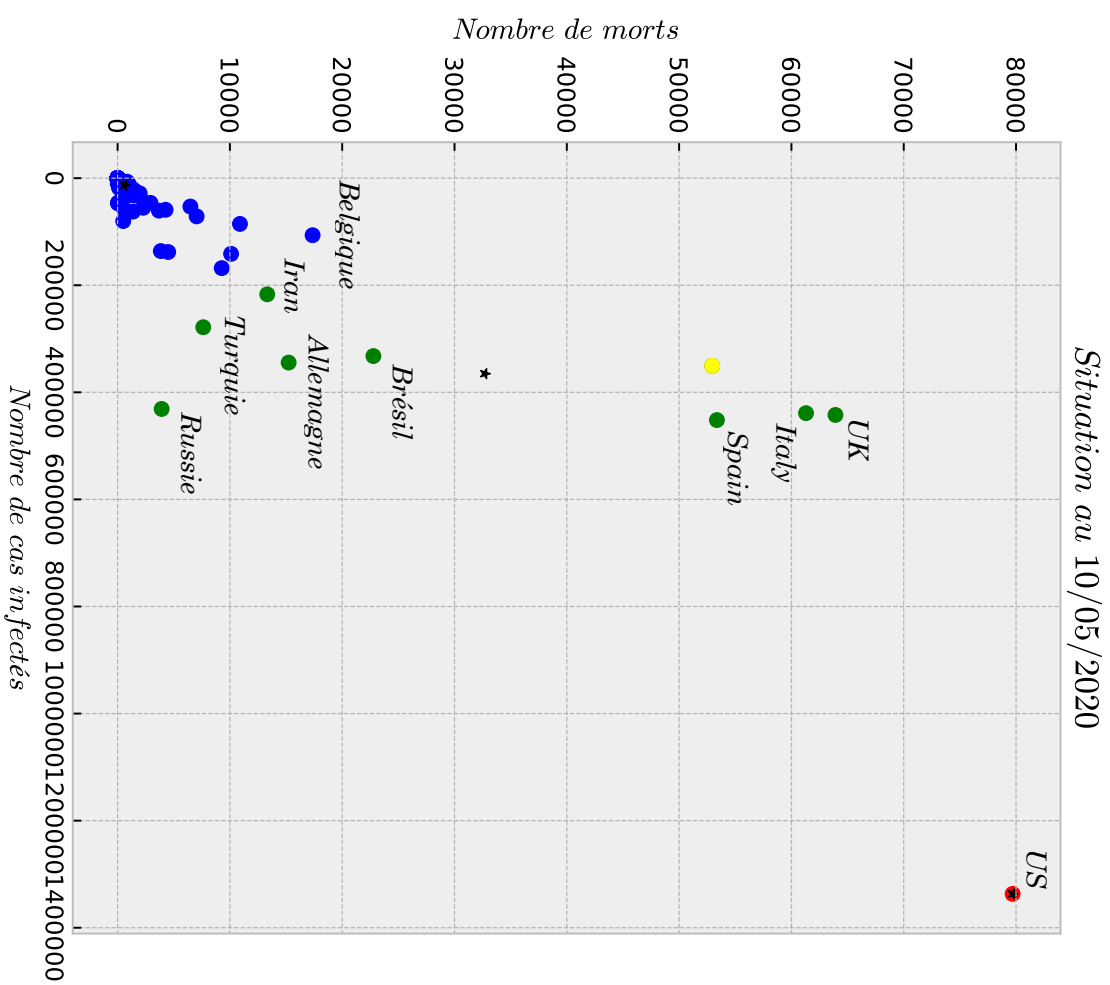
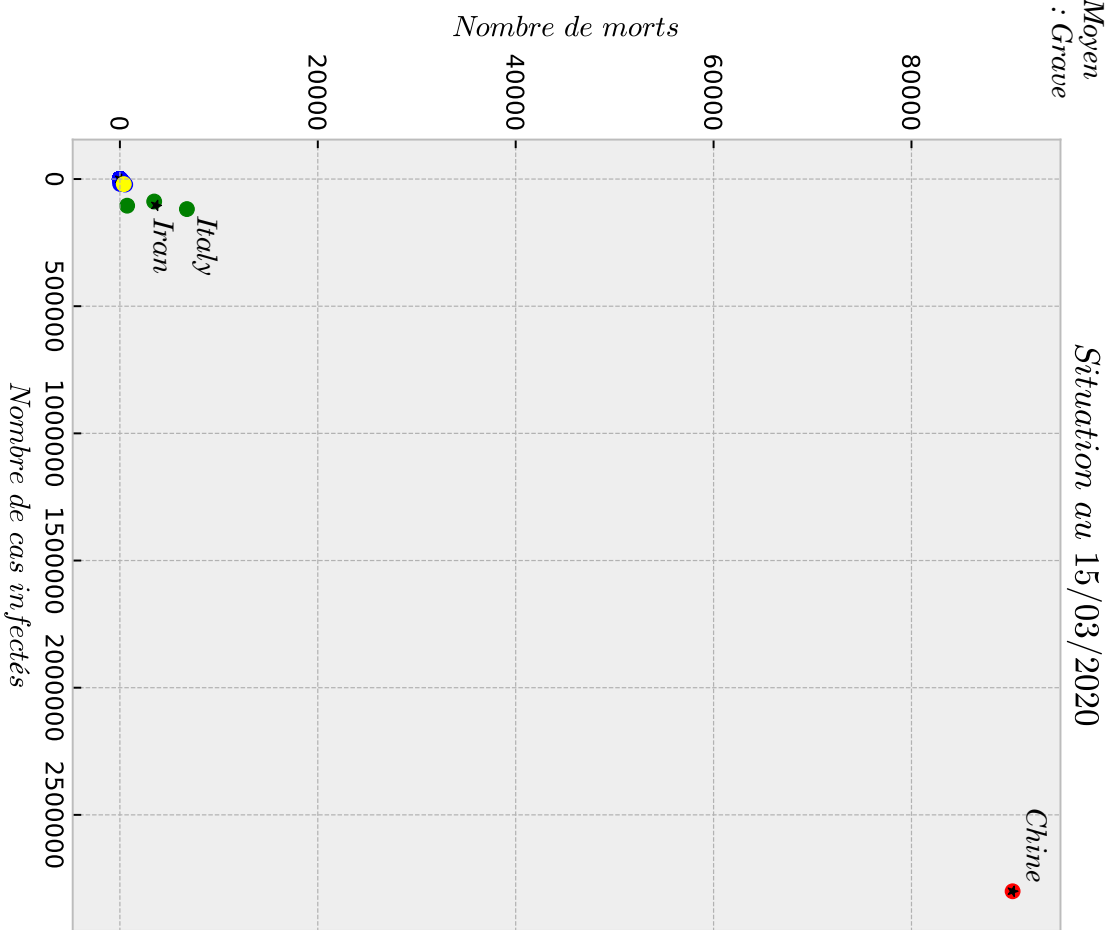
- Infectés : 350 506
- Morts : 52 945

La figure suivante nous montre l'exécution du programme dans la console :

```
CENTROIDS LOCATIONS :  
[[8.14158470e+02 1.08087432e+01]  
 [2.80042600e+06 9.02210000e+04]  
 [1.03745000e+05 3.65733333e+03]]  
*****  
PREDICTION FOR FRANCE (15/03/2020) :  
Infected : 4501  
Dead : 91  
Cluster : blue  
*****  
*****  
PREDICTION FOR FRANCE (10/05/2020) :  
Infected : 176970  
Dead : 26380  
Cluster : green  
*****
```

Nos résultats sous forme graphique sont accessible sur la page suivante. Vous trouverez aussi l'intégralité du code *Python* à la fin de cette présentation

- Jaune : Situation de la France
- Bleu : Faible
- Vert : Moyen
- Rouge : Grave



Interprétation des résultats :

On observe que la France se place, avant le confinement, dans les pays faiblement touchés (en bleu), on recense quelques cas mais rien de grave. Du fait d'un phénomène de latence dans le temps, la situation de la France c'est aggravée pendant le confinement et celle-ci rejoint les pays moyennement touchés par l'épidémie. Elle fait partie des pays les plus touchés dans le monde avec l'Italie, l'Espagne et le Royaume-Unis.

A l'heure actuelle, on observe que les Etats-Unis se place en secteur rouge alors qu'au 15/03 celui-ci se plaçait en secteur bleu. C'est le pays le plus largement atteint par l'épidémie au vu du nombre de morts et du nombre de cas infectés extrêmement élevé par rapport aux autres pays. A l'inverse, à la date du 15/03, la Chine était le pays le plus touché. Mais à la suite de mesures de confinements drastiques, elle a pu enrayer l'épidémie. Comme le montre notre graphe à l'heure actuelle la Chine est en secteur bleu.

Dans notre étude, nous utilisons les valeurs cumulées dans le temps du nombre de morts et du nombre de cas infectés depuis le début de l'épidémie. Il aurait été intéressant d'obtenir les valeurs journalières pour les deux dates, c'est à dire le nombre de nouveau mort et le nombre de nouveau cas infectés aux dates du 15/03 et du 10/05 pour chaque pays, et enfin en observer les conséquences. Nous n'avons pas réussi à obtenir ces valeurs journalières par manque de temps, néanmoins pour pallier ce problème, nous avons calculé, pour chaque pays, le facteur d'évolution. Ce facteur d'évolution nous donne une indication de l'évolution de l'épidémie dans chaque pays. En effet, une valeur égale à 1 du facteur d'évolution correspond à une situation de stagnation, une valeur supérieure à 1 équivaut à un accroissement de l'épidémie et à l'inverse, équivaut à une déclinaison de l'épidémie.

Par exemple pour la France au 15/03, le facteur d'évolution était de 1,48 et au 10/05 celui-ci était de 1,15. On voit bien que les mesures de confinement ont eu un impact sur l'évolution de l'épidémie mais cette valeur reste supérieure à 1. Donc l'heure est à la prudence, la tendance est à la baisse mais des efforts seront encore nécessaires pour ramener le facteur d'évolution en dessous de 1. On peut donner d'autres exemples :

Au 15/03 :

- Etats-Unis : 1,30
- Chine : 2,01

Au 10/05 :

- Etats-Unis : 0,82
- Chine : 0,74

```

1 from cirilib.imports import *
2 # from data_extracion import *
3 from sklearn.cluster import KMeans
4 from matplotlib import style
5
6 sns.set(style="whitegrid")
7 style.use("bmh")
8
9 lockdown_date = "2020-03-15"
10 # Let k assume a value
11 k = 3
12
13
14 def clustering(lockdown_date, k=3, backend="sns"):
15     before_after = ["$before$ ", "$after$ "]
16
17     try:
18         df1 = pd.read_csv(os.path.join(CSV_DIR, "before_" +
19 lockdown_date))
20         df2 = pd.read_csv(os.path.join(CSV_DIR, "after_" + lockdown_date
21 ))
22         df_before_after = (df1, df2)
23         print("\33[31m", "The data has been extracted from already
24 existing files, consider regenerating them!", "\33[0m")
25     except FileNotFoundError:
26         df_before_after = lockdown_split(lockdown_date, to_csv=True)
27
28     fig, ax = plt.subplots(ncols=2)
29     fig.subplots_adjust(hspace=0.5)
30
31     for i in range(2):
32         df = df_before_after[i]
33
34         print("DATABASE OVERVIEW :")
35         print(df.head)
36
37         # Create an instance of the KMeans class with a cluster size of
38         k
39         kmeans = KMeans(n_clusters=k)
40
41         # Create a matrix containing all points
42         Countries = list(df['Country'])
43         X = np.array(list(zip(df['Cases'], df['Deaths'])))
44
45         # Train the model
46         kmeans = kmeans.fit(X)
47
48         # Assign a label to all of the points
49         labels = kmeans.predict(X)
50
51         # Get the centroids
52         Centroids = kmeans.cluster_centers_
53
54         # Print the clusters label and centroids
55         print("LABELS :")
56         print(labels)
57         print("CENTROIDS LOCATIONS :")

```

```

54     print(Centroids)
55
56     flatui = ["#9b59b6", "#3498db", "#95a5a6", "#e74c3c", "#34495e",
57             "#2ecc71"]
58
59     if backend == "plt":
60         # Plot the points and centroids on a scatter plot
61         # And map the labels to colors
62
63         colors = [flatui[i] for i in labels]
64
65         ax[i].set(title=("$State$ " + before_after[i] +
66             lockdown_date), xlabel="$Confirmed$ $Cases$",
67             ylabel="$Reported$ $Deaths$")
68
69         ax[i].scatter(df['Cases'], df['Deaths'], c=colors)
70         ax[i].scatter(Centroids[:, 0], Centroids[:, 1], marker='*',
71             c='black', s=20)
72
73     else:
74         clarity_ranking = COUNTRIES
75         sns.scatterplot(x="Cases", y="Deaths", hue="Country",
76             palette="ch:r=-.2,d=.3_r", size="Growth Factor",
77             hue_order=clarity_ranking, data=df, ax=ax[i
78         ]).legend_.remove()
79
80         # Print out the clusters to which point belongs
81         print("CLUSTERS TO WHICH POINT BELONGS :")
82         for i, labels in enumerate(labels):
83             print(Countries[i] + str(X[i]), "Cluster " + str(int(labels
84         )))
85
86         # Before was the training part
87         # Testing part for FRANCE
88         # Making predictions
89         print("*****")
90         Cases = 4501
91         Deaths = 91
92         print("PREDICTION FOR FRANCE before", lockdown_date, ":")
93         print("Infected : 4501")
94         print("Dead : 91")
95         cluster = kmeans.predict([[Cases, Deaths]])[0]
96         plt.scatter(Cases, Deaths, c='#fac205')
97         print("Cluster : ", flatui[cluster])
98         print("*****")
99
100     plt.show()
101     return
102
103 if __name__ == "__main__":
104     clustering(lockdown_date)

```