Projet Forage de Données

**Les données :**

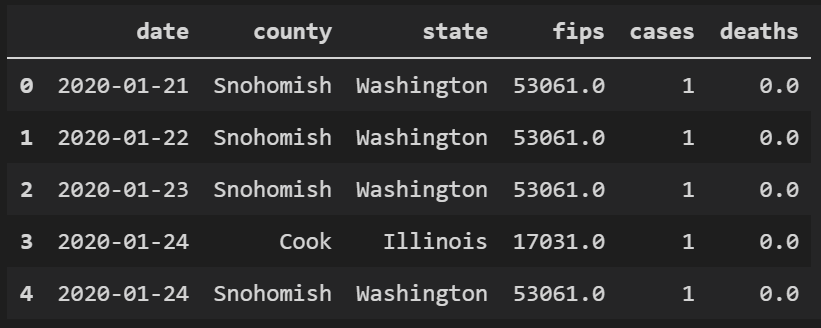
Nous avons utilisé la dataset publique suivante, trouvée sur :

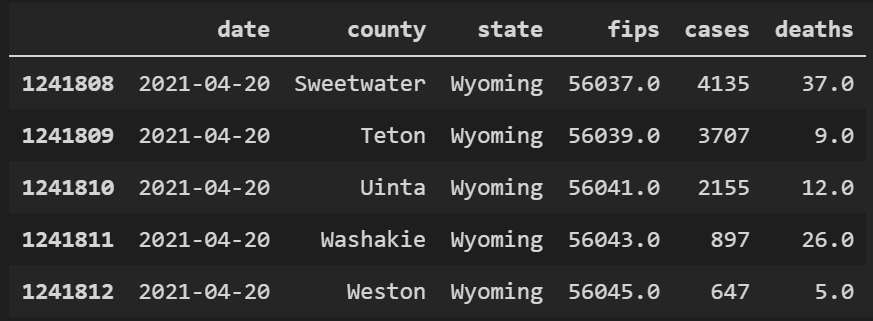
<https://www.kaggle.com/fireballbyedimyrnmom/us-counties-covid-19-dataset>

Cette dataset concerne le nombre de cas quotidiens des personnes atteintes par la Covid19 dans les Etats-Unis sainsi que le nombre quotidien des décès et ce regroupes par comté dans tout le pays.

Nous avons pensé alors que ça serait un bon exemple pour démontrer l’utilité de la prédiction des séries temporelles par des méthodes comme ARIMA et LSTM, et comment cela peut aider dans des situations de la vie réelle.

|  |
| --- |
| df = pd.read\_csv('us-counties.csv')  df.head()  df.tail() |



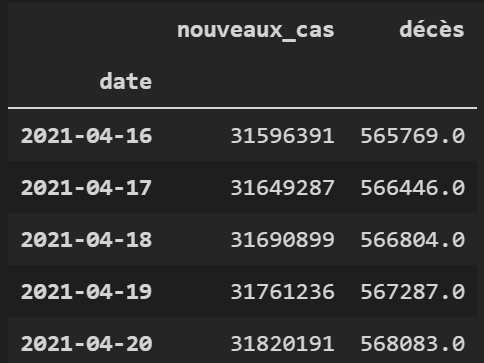


La dataset contient jusqu’à 1,241,812 entrées depuis le 21 janvier 2020 et jusqu’au 20 avril 2021.

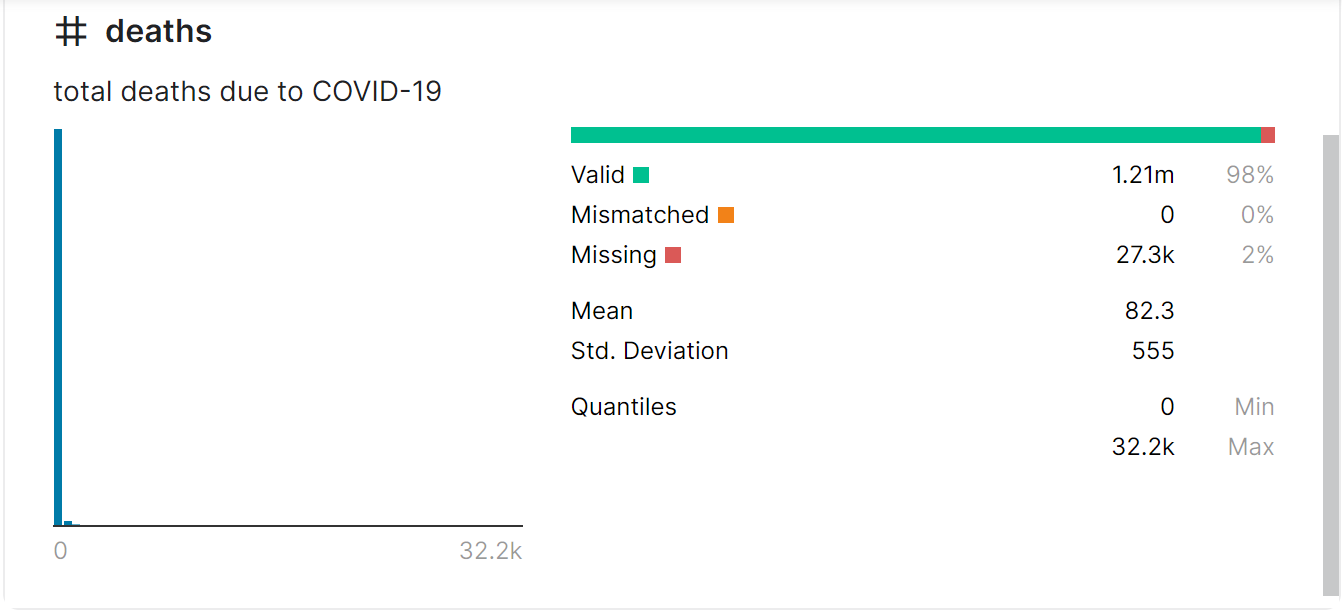
**Préparation des données :**

* Les données sont regroupées par chaque comté, ce qu’on n’en aura pas besoin, le plus important pour nous est d’avoir le nombre de cas de chaque jour dans tous les Etats-Unis. On a alors additionné le nombre de cas de chaque jour, en ne conservant que la colonne ‘date’, ‘nouveaux\_cas’ et ‘décès’.

|  |
| --- |
| #additionner les valeurs (cas et décès) de chaque date  df = df.groupby('date').sum()  # éliminer la colonne fips  df = df.drop("fips", axis= 1)  df.columns = ['nouveaux\_cas', 'décès']  df.head()  df.tail() |

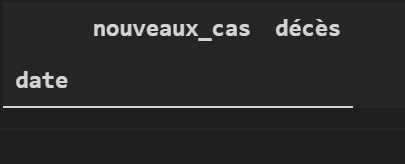
 

* Pour les cas de décès, il y a 27,3 valeurs manquantes, ce qu’on peut faire tout simplement est de les remplir avec des zéros.



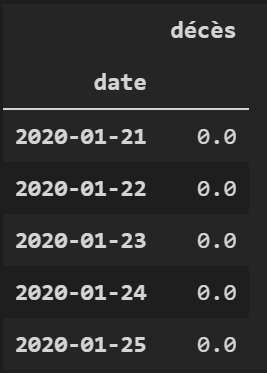
Mais apparemment après l’addition des valeurs de chaque date, on n’a plus de valeurs manquantes, comme affiché ci-dessous, ce qui implique que pandas a remplacé automatiquement ces valeurs par 0.

|  |
| --- |
| df.query("décès == ‘’ ") |



* Ensuite, on a divisé les données en deux, données pour les nouveaux cas, et données pour les cas de décès de chaque jour, pour prédire juste un à la fois.

|  |
| --- |
| df\_cas = pd.DataFrame(df['nouveaux\_cas'])  df\_deces = df\_cas = pd.DataFrame(df['décès'])  df\_cas.head()  df\_deces.head() |



**La finalité :**

Notre objectif est de prédire les nouveaux cas enregistrés du Coronavirus en Etats-Unis ainsi que les nouveaux cas de décès.

**Les series temporelles :**

**Composants :**

* **Niveau :** la ligne de base d’une serie temporelle si elle était une ligne droite
* **Tendance :** l’augumentation ou la dimunition des series temporelles en fonction du temps souvent lineaire
* **Saisonnalite :** le comportement repetitifs sur des cycles du temps des series temporelles
* **Bruit :** la variabilité aleatoire des series temporelles qui ne peut pas etre expliquee par le modele.

On peut modéliser n’importe quelle serie temporelle en tant que :

|  |
| --- |
| **Y = niveau + tendance + saisonnalite + bruit** |

**ARIMA :**

ARIMA ou « **A**uto **R**egressive **I**ntegrated **M**oving **A**verage » est un modele statistique très simple, qui peut être represente par la relation suivante :

|  |
| --- |
| y[t+1] = (c + a0\*y[t] + a1\*y[t-1] +…+ ap\*y[t-p]) +  (e[t] + b1\*e[t-1] + b2\*e[t-2] +…+ bq\*e[t-q]) |

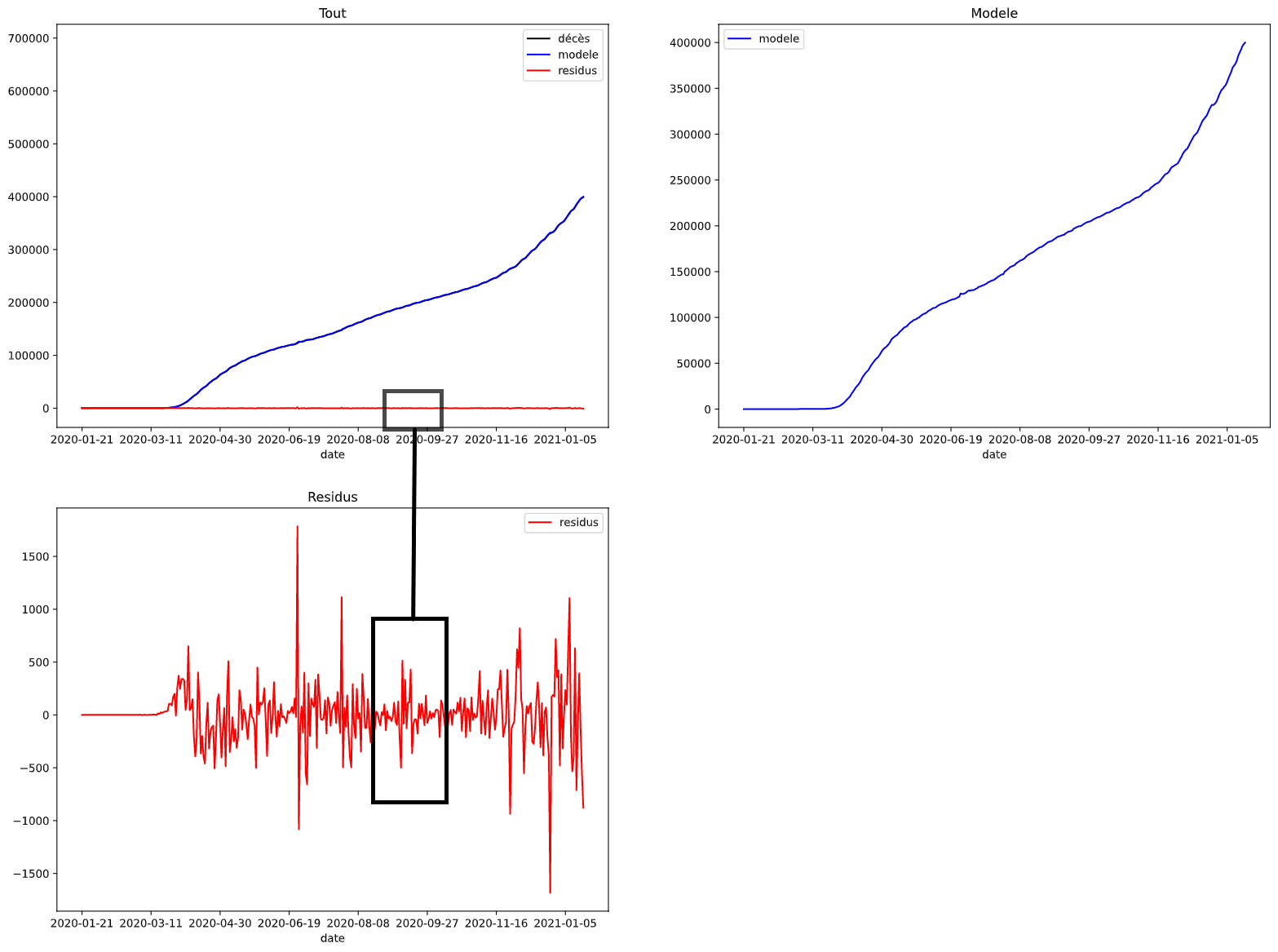
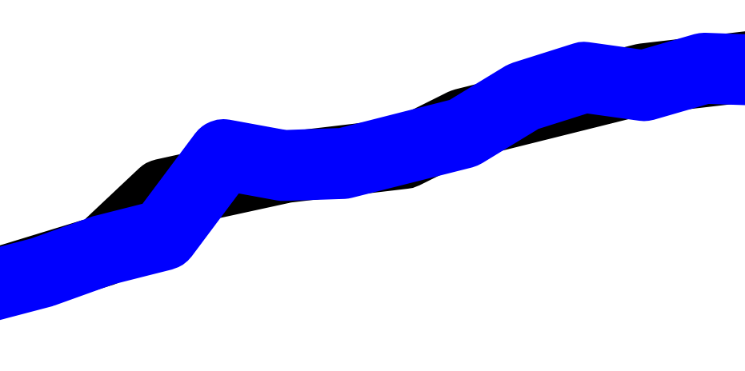
et comme son nom l’indique, ARIMA se caracterise par trois caractéristiques representees par trois composantes comme montree dans la relation ci-dessus, c’est un modele :

* Autorégressif : La regression s’effectue par rapport aux valeurs passees de y, cette caracteristique est representee par la composante **p** ou **AR** : le nombre des coefficients pi .
* Moyenne mobile (Moving average) : **q** ou **MA** elimination du mouvement aleatoire des series temporelles ou ce qu’on appelle le bruit
* Integree (Integrated) : c’est une propriete qui permet d’eliminer la tendance et la saisonnalité des series temporelles, et est representee par la composante **d** qui signifie le nombre de fois qu’il faut différencier y pour atteindre l’integralite.

Cette dernière propriete nous permet d’avoir la stationnarite de notre modele i.e :

* Les modeles purement autoregressif..
* Les complications d’ajouter une erreur…hill climbing…
* Trouver les composantes (p,d,q)
* Les composantes de la saisonnalite…
* Les bibliothèques existantes..

Le nombre des nouveaux cas :



**LSTM :**

Références :

[https://machinelearningmastery.com/time-series-forecasting](https://machinelearningmastery.com/time-series-forecasting/#:~:text=Making%20predictions%20about%20the%20future,them%20to%20predict%20future%20observations)

<https://medium.com/analytics-vidhya/time-series-forecasting-arima-vs-lstm-vs-prophet-62241c203a3b>

<https://people.duke.edu/~rnau/411arim.htm>

<https://www.analyticsvidhya.com/blog/2020/10/how-to-create-an-arima-model-for-time-series-forecasting-in-python/>