

Master bio-informatique et modélisation des systèmes complexes appliquée à la Santé

VISUALISATION, PRÉDICTION DE L'ÉVOLUTION DE COVID-19 AU MAROC

Projet de Machine Learning

RÉALISÉ PAR :

Aicha Laamarti

Afkir Youssef

ENCADRÉ PAR :

Pr. Abdellah Mahmoudi

BIO-MSCS 2019/2020

Contents

INTRODUCTION GÉNÉRALE	3
CHAPITRE I : DEFENITIONS	4
1. Coronavirus (covid-19)	5
1.1 Symptômes	5
1.2 Causes	7
1.3 Complications	7
1.4 Prévention.....	7
2. Machine Learning	8
1.1 Types d'apprentissage automatique	9
CHAPITRE II : RÉALISATION DU PROJET	11
1. Outils, langage et Bibliothèques utilisés	12
2. Base de données.....	13
3. Analyse de la situation actuelle au Maroc.....	13
4. Prévision du nombre total de cas - décès et cas confirmés.....	24
1.1 Prédéterminer la tendance des cas confirmés dans les 10 prochains jours.....	27
1.2 Prédire les nouveaux cas quotidiens.....	30
1.3 Prédire le nombre total de décès	32
CONCLUSION	33

INTRODUCTION GÉNÉRALE

La maladie à coronavirus (COVID-19) est une maladie infectieuse causée par un coronavirus nouvellement découvert. Le premier cas de coronavirus a été appréhendé dans la ville de Wuhan, en Chine, à la fin de décembre 2019. Avec l'Organisation mondiale de la Santé déclarant une pandémie. Le taux de décès est d'environ 2,3 % (selon les renseignements initiaux). De nombreux pays dans le monde ont reconnu la situation dangereuse de ce nouveau virus, et ils ont donc commencé à appliquer de nombreuses procédures pour ralentir sa propagation rapide entre les gens, comme le confinement et la distanciation sociale.

Le premier cas, qui provenait en fait de l'Italie, a été identifié au Maroc le 2 mars 2020. Deux semaines plus tard, les autorités marocaines ont commencé à prendre des mesures préventives pour ralentir l'épidémie. L'ensemble de données sur lequel nous travaillons a été enregistré entre le 2 mars 2020 et le 26 août 2020.

CHAPITRE I : DEFENITIONS

1. Coronavirus (covid-19)

Les coronavirus sont une famille de virus qui peuvent causer des maladies comme le rhume, le syndrome respiratoire aigu sévère (SRAS) et le syndrome respiratoire du Moyen-Orient (SRMO). En 2019, un nouveau coronavirus a été identifié comme étant la cause d'une éclosion de maladie qui a pris naissance en Chine. Le virus est maintenant connu sous le nom de coronavirus 2 du syndrome respiratoire aigu sévère (SRAS-CoV-2). La maladie qu'elle cause est appelée coronavirus 2019 (COVID-19). En mars 2020, l'Organisation mondiale de la Santé (OMS) a déclaré que l'éclosion de COVID-19 était une pandémie. Des groupes de santé publique, y compris les Centers for Disease Control and Prevention (CDC) des États-Unis et l'OMS, surveillent la pandémie et publient des mises à jour sur leurs sites Web. Ces groupes ont également émis des recommandations pour prévenir et traiter la maladie.

1.1 Symptômes

Les signes et symptômes de la maladie à coronavirus 2019 (COVID-19) peuvent apparaître deux à 14 jours après l'exposition. Cette période après l'exposition et avant d'avoir des symptômes est appelée la période d'incubation. Les signes et symptômes communs peuvent inclure :

- Fièvre
- Toux
- Fatigue.
- Les premiers symptômes de la COVID-19 peuvent inclure une perte de goût ou d'odeur

D'autres symptômes peuvent inclure :

- Essoufflement ou difficulté à respirer
- Douleurs musculaires
- Frissons
- Mal de gorge
- Nez qui coule
- Mal de tête
- Douleur thoracique
- Œil rose (conjonctivite)

Cette liste n'est pas exhaustive. D'autres symptômes moins courants ont été signalés, comme des éruptions, des nausées, des vomissements et de la diarrhée. Les enfants

présentent des symptômes semblables à ceux des adultes et sont généralement atteints d'une maladie bénigne.

La gravité des symptômes de la COVID-19 peut varier de très légère à grave. Certaines personnes peuvent avoir seulement quelques symptômes, et certaines personnes peuvent n'avoir aucun symptôme du tout. Certaines personnes peuvent présenter des symptômes aggravés, comme un essoufflement et une pneumonie aggravée, environ une semaine après le début des symptômes.

Les personnes plus âgées sont plus à risque de contracter une maladie grave à cause de la COVID-19, et le risque augmente avec l'âge. Les personnes qui ont déjà des problèmes de santé chroniques peuvent aussi présenter un risque plus élevé de maladie grave. Les problèmes de santé graves qui augmentent le risque de maladie grave attribuable à la COVID-19 comprennent :

- Maladies cardiaques graves, comme l'insuffisance cardiaque, la coronaropathie ou la cardiomyopathie
- Cancer
- Maladie pulmonaire obstructive chronique (MPOC)
- Diabète de type 2
- Obésité grave
- Maladie rénale chronique
- Drépanocytose
- Système immunitaire affaibli par les greffes d'organes solides

D'autres conditions peuvent accroître le risque de maladie grave, comme :

- Asthme
- Maladie du foie
- Maladies pulmonaires chroniques comme la mucoviscidose
- Troubles du cerveau et du système nerveux
- Système immunitaire affaibli par la greffe de moelle osseuse, le VIH ou certains médicaments
- Diabète de type 1
- Hypertension artérielle

Cette liste n'est pas exhaustive. D'autres problèmes médicaux sous-jacents peuvent accroître votre risque de maladie grave attribuable à la COVID-19.

1.2 Causes

L'infection par le nouveau coronavirus (coronavirus 2 du syndrome respiratoire aigu sévère, ou SRAS-CoV-2) cause la maladie à coronavirus 2019 (COVID-19). Le virus semble se propager facilement parmi les gens, et d'autres continuent d'être découverts au fil du temps sur la façon dont il se propage. Les données ont montré qu'il se propage d'une personne à l'autre parmi les personnes en contact étroit (à environ 6 pieds, ou 2 mètres). Le virus se propage par des gouttelettes respiratoires libérées lorsqu'une personne infectée tousse, éternue ou parle. Ces gouttelettes peuvent être inhalées ou atterrir dans la bouche ou le nez d'une personne à proximité. Il peut également se propager si une personne touche une surface avec le virus dessus et touche ensuite sa bouche, son nez ou ses yeux, bien que ce ne soit pas considéré comme un moyen principal il se propage.

1.3 Complications

Bien que la plupart des personnes atteintes de la COVID-19 présentent des symptômes légers à modérés, la maladie peut entraîner des complications médicales graves et entraîner la mort chez certaines personnes. Les adultes plus âgés ou les personnes qui ont déjà des problèmes de santé chroniques courent un plus grand risque de devenir gravement malades à cause de la COVID-19.

Les complications peuvent comprendre :

- Pneumonie et difficulté à respirer
- Défaillance d'organes dans plusieurs organes
- Problèmes cardiaques
- Affection pulmonaire grave qui fait passer une faible quantité d'oxygène dans votre circulation sanguine jusqu'à vos organes (syndrome de détresse respiratoire aiguë)
- Caillots de sang
- Lésion rénale aiguë
- Autres infections virales et bactériennes

1.4 Prévention

Bien qu'aucun vaccin ne soit disponible pour prévenir la COVID-19, vous pouvez prendre des mesures pour réduire votre risque d'infection. L'OMS et les CDC recommandent de prendre les précautions suivantes pour éviter la COVID-19 :

- Évitez les grands événements et les rassemblements de masse.

- Éviter tout contact étroit (à moins de 6 pieds ou 2 mètres) avec une personne malade ou présentant des symptômes.
- Restez à la maison autant que possible et gardez une distance entre vous et les autres (à environ 6 pieds, ou 2 mètres), surtout si vous avez un risque plus élevé de maladie grave. N'oubliez pas que certaines personnes peuvent avoir la COVID-19 et la transmettre à d'autres, même si elles n'ont pas de symptômes ou ne savent pas qu'elles ont la COVID-19.
- Lavez-vous souvent les mains avec de l'eau et du savon pendant au moins 20 secondes ou utilisez un désinfectant pour les mains à base d'alcool qui contient au moins 60 % d'alcool.
- Couvrez votre visage avec un masque en tissu dans les espaces publics, comme l'épicerie, où il est difficile d'éviter le contact étroit avec les autres, surtout si vous vous trouvez dans une zone où la propagation de la communauté se poursuit. Utiliser uniquement des masques en tissu non médicaux — les masques chirurgicaux et les appareils respiratoires N95 devraient être réservés aux fournisseurs de soins de santé.
- Couvrez votre bouche et votre nez avec votre coude ou un mouchoir lorsque vous toussiez ou éternuez. Jetez le mouchoir utilisé. Lavez-vous les mains immédiatement.
- Évitez de toucher vos yeux, votre nez et votre bouche.
- Évitez de partager la vaisselle, les verres, les serviettes, la literie et d'autres articles ménagers si vous êtes malade.
- Nettoyez et désinfectez les surfaces fréquemment touchées, comme les poignées de porte, les interrupteurs d'éclairage, les appareils électroniques et les comptoirs, tous les jours.
- Si vous êtes malade, restez à la maison, loin du travail, de l'école et des lieux publics, à moins d'obtenir des soins médicaux. Évitez les transports en commun, les taxis et le covoiturage si vous êtes malade.
- Si vous avez un problème de santé chronique et que vous risquez davantage de contracter une maladie grave, consultez votre médecin pour savoir comment vous protéger.

2. Machine Learning

L'apprentissage automatique est une application de l'intelligence artificielle (IA) qui fournit aux systèmes la capacité d'apprendre et d'améliorer automatiquement à partir de l'expérience sans être explicitement programmé. L'apprentissage

automatique se concentre sur le développement de programmes informatiques qui peuvent accéder aux données et les utiliser pour apprendre par eux-mêmes.

Le processus d'apprentissage commence par des observations ou des données, comme des exemples, de l'expérience directe ou de l'enseignement, afin de rechercher des tendances dans les données et de prendre de meilleures décisions à l'avenir en fonction des exemples que nous fournissons. L'objectif principal est de permettre aux ordinateurs d'apprendre automatiquement sans intervention ni assistance humaine et d'ajuster les actions en conséquence.

Mais, en utilisant les algorithmes classiques de l'apprentissage automatique, le texte est considéré comme une séquence de mots clés ; au lieu de cela, une approche basée sur l'analyse sémantique imite la capacité humaine à comprendre le sens d'un texte.

1.1 Types d'apprentissage automatique

L'apprentissage automatique classique est souvent classé par la façon dont un algorithme apprend à devenir plus précis dans ses prédictions. Il existe quatre approches de base : l'apprentissage supervisé, l'apprentissage non supervisé, l'apprentissage semi-supervisé et l'apprentissage de renforcement. Le type d'algorithme qu'un scientifique choisit d'utiliser dépend du type de données qu'il veut prédire.

- **Apprentissage supervisé** : Dans ce type d'apprentissage automatique, les data scientists fournissent des algorithmes avec des données de formation étiquetées et définissent les variables qu'ils veulent que l'algorithme évalue pour les corrélations. L'entrée et la sortie de l'algorithme sont spécifiées.
- **Apprentissage non supervisé** : Ce type d'apprentissage machine implique des algorithmes qui s'entraînent sur des données non étiquetées. L'algorithme parcourt des ensembles de données à la recherche d'une connexion significative. Les algorithmes de données s'entraînent et les prédictions ou recommandations qu'ils produisent sont prédéterminées.
- **Apprentissage semi-supervisé** : Cette approche de l'apprentissage automatique comporte un mélange des deux types précédents. Les data scientists peuvent alimenter un algorithme principalement étiqueté données de formation, mais le modèle est libre d'explorer les données sur son propre et de développer sa propre compréhension de l'ensemble de données.

- **Renforcement de l'apprentissage :** L'apprentissage du renforcement est généralement utilisé pour enseigner à une machine à compléter un processus en plusieurs étapes pour lequel il existe des règles clairement définies. Les scientifiques des données programment un algorithme pour accomplir une tâche et lui donnent des indices positifs ou négatifs pendant qu'il travaille sur la façon d'accomplir une tâche. Mais dans la plupart des cas, l'algorithme décide seul des étapes à suivre.

CHAPITRE II : RÉALISATION DU PROJET

1. Outils, langage et Bibliothèques utilisés

- **Python** est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Parce que sa syntaxe est élégante, que son typage est dynamique et qu'il est interprété, Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes.
- **Jupyter** est un environnement de développement interactif basé sur le Web pour les ordinateurs portables Jupyter, le code et les données. Jupyter est flexible : configurer et organiser l'interface utilisateur pour prendre en charge un large éventail de flux de travail dans la science des données, l'informatique scientifique et l'apprentissage automatique. Jupyter est extensible et modulaire : des plugins d'écriture qui ajoutent de nouveaux composants et s'intègrent aux composants existants.
- **NumPy** est une extension du langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.
- **Pandas** est une bibliothèque écrite pour le langage de programmation Python permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles. Pandas est un logiciel libre sous licence BSD2.
- **Matplotlib** est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques⁵. Elle peut être combinée avec les bibliothèques python de calcul scientifique NumPy et SciPy⁶. Matplotlib est distribuée librement et gratuitement sous une licence de style BSD4. Sa version stable actuelle (la 2.0.1 en 2017) est compatible avec la version 3 de Python.
- **Seaborn** est une bibliothèque pour créer des graphiques statistiques en Python. Elle s'appuie sur matplotlib et s'intègre étroitement avec les structures de données pandas.

- **Prophet** est une bibliothèque open-source conçue pour faire des prévisions pour des ensembles de données de séries chronologiques univariées. Il est facile à utiliser et conçu pour trouver automatiquement un bon ensemble d'hyperparamètres pour le modèle dans un effort de faire des prévisions habiles pour les données avec des tendances et la structure saisonnière par défaut.
- **Plotly** est une plateforme graphique et analytique collaborative basée sur le Web.

2. Base de données

Cet ensemble de données contient des données quotidiennes sur la situation de la COVID-19 au Maroc.

	Date	Confirmed	Deaths	Recovered	Excluded	Beni Mellal-Khenifra	Casablanca-Settat	Draa-Tafilalet	Dakhla-Oued Ed-Dahab	Fes-Meknes	Guelmim-Oued Noun	Laayoune-Sakia El Hamra	Marrakesh-Safi	Oriental	Rabat-Sale-Kenitra
0	2020-03-02	1.0	NaN	NaN	28.0	NaN	1.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	2020-03-03	NaN	NaN	NaN	32.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	2020-03-04	2.0	NaN	NaN	34.0	NaN	2.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	2020-03-05	NaN	NaN	NaN	40.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	2020-03-06	NaN	NaN	NaN	50.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...
173	2020-08-22	50812.0	858.0	35040.0	1686164.0	1982.0	14035.0	1809.0	275.0	7790.0	224.0	961.0	8257.0	920.0	4665.0
174	2020-08-23	52349.0	888.0	36343.0	1706521.0	2104.0	14673.0	1884.0	277.0	8019.0	235.0	973.0	8428.0	931.0	4768.0
175	2020-08-24	53252.0	920.0	37478.0	1726728.0	2204.0	14963.0	1917.0	287.0	8158.0	234.0	974.0	8589.0	958.0	4845.0
176	2020-08-25	54528.0	955.0	38293.0	1747642.0	2345.0	15361.0	2029.0	311.0	8300.0	235.0	982.0	8763.0	1004.0	4968.0
177	2020-08-26	55864.0	984.0	40586.0	1768387.0	2453.0	15911.0	2067.0	325.0	8425.0	235.0	995.0	8967.0	1051.0	5084.0

178 rows x 17 columns

L'ensemble de données comprend 16 colonnes : date, cas confirmés, décès, rétablissement et nombre de cas dans chacune des 12 régions marocaines.

```
Index(['Date', 'Confirmed', 'Deaths', 'Recovered', 'Excluded',
      'Beni Mellal-Khenifra', 'Casablanca-Settat', 'Draa-Tafilalet',
      'Dakhla-Oued Ed-Dahab', 'Fes-Meknes', 'Guelmim-Oued Noun',
      'Laayoune-Sakia El Hamra', 'Marrakesh-Safi', 'Oriental',
      'Rabat-Sale-Kenitra', 'Souss-Massa', 'Tanger-Tetouan-Al Hoceima'],
      dtype='object')
```

3. Analyse de la situation actuelle au Maroc

- Importation des bibliothèques :

```

import numpy as np
import pandas as pd
pd.plotting.register_matplotlib_converters()
import matplotlib.pyplot as plt
import plotly.express as px
#%%matplotlib inline
import seaborn as sns
import datetime as dt
from fbprophet import Prophet
import plotly.express as px
import plotly.graph_objects as go

```

- Importer les données :

```
my_data = pd.read_excel('C:/Users/youuss/OneDrive/Bureau/corona_morocco.xlsx')
```

- Convertir « Date column » en « datetime » pour une meilleure visualisation :

```
my_data['Date']=[dt.datetime.strptime(x,'%d/%m/%Y') for x in my_data['Date'] ]
```

- Informations de base de données :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 17 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Date                                178 non-null    datetime64[ns]
1   Confirmed                          172 non-null    float64
2   Deaths                            161 non-null    float64
3   Recovered                          160 non-null    float64
4   Excluded                           169 non-null    float64
5   Beni Mellal-Khenifra              158 non-null    float64
6   Casablanca-Settat                 161 non-null    float64
7   Draa-Tafilalet                    158 non-null    float64
8   Dakhla-Oued Ed-Dahab              158 non-null    float64
9   Fes-Meknes                         159 non-null    float64
10  Guelmim-Oued Noun                 158 non-null    float64
11  Laayoune-Sakia El Hamra           158 non-null    float64
12  Marrakesh-Safi                    160 non-null    float64
13  Oriental                           158 non-null    float64
14  Rabat-Sale-Kenitra                158 non-null    float64
15  Souss-Massa                       158 non-null    float64
16  Tanger-Tetouan-Al Hoceima         158 non-null    float64
dtypes: datetime64[ns](1), float64(16)
memory usage: 23.8 KB
```

NB : On remarque qu'on a plusieurs valeurs nulles

- Remplacement des valeurs nulles par la valeur 0 :

```
my_data = my_data.fillna(0)
```

Maintenant toutes les valeurs NULL sont mises à zéro :

Avant :

	Date	Confirmed	Deaths	Recovered	Excluded	Beni Mellal-Khenifra	Casablanca-Settat	Draa-Tafilalet	Dakhla-Oued Ed-Dahab	Fes-Meknes	Guelmim-Oued Noun	Laayoune-Sakia El Hamra	Marrakesh-Safi	Oriental	Rabat-Sale-Kenitra
0	2020-03-02	1.0	NaN	NaN	28.0	NaN	1.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	2020-03-03	NaN	NaN	NaN	32.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	2020-03-04	2.0	NaN	NaN	34.0	NaN	2.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	2020-03-05	NaN	NaN	NaN	40.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	2020-03-06	NaN	NaN	NaN	50.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Après :

	Date	Confirmed	Deaths	Recovered	Excluded	Beni Mellal-Khenifra	Casablanca-Settat	Draa-Tafilalet	Dakhla-Oued Ed-Dahab	Fes-Meknes	Guelmim-Oued Noun	Laayoune-Sakia El Hamra	Marrakesh-Safi	Oriental	Rabat-Sale-Kenitra
0	2020-03-02	1.0	0.0	0.0	28.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	2020-03-03	0.0	0.0	0.0	32.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	2020-03-04	2.0	0.0	0.0	34.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	2020-03-05	0.0	0.0	0.0	40.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	2020-03-06	0.0	0.0	0.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

- Définir la colonne Date comme index

```
my_data = my_data.set_index('Date')
```

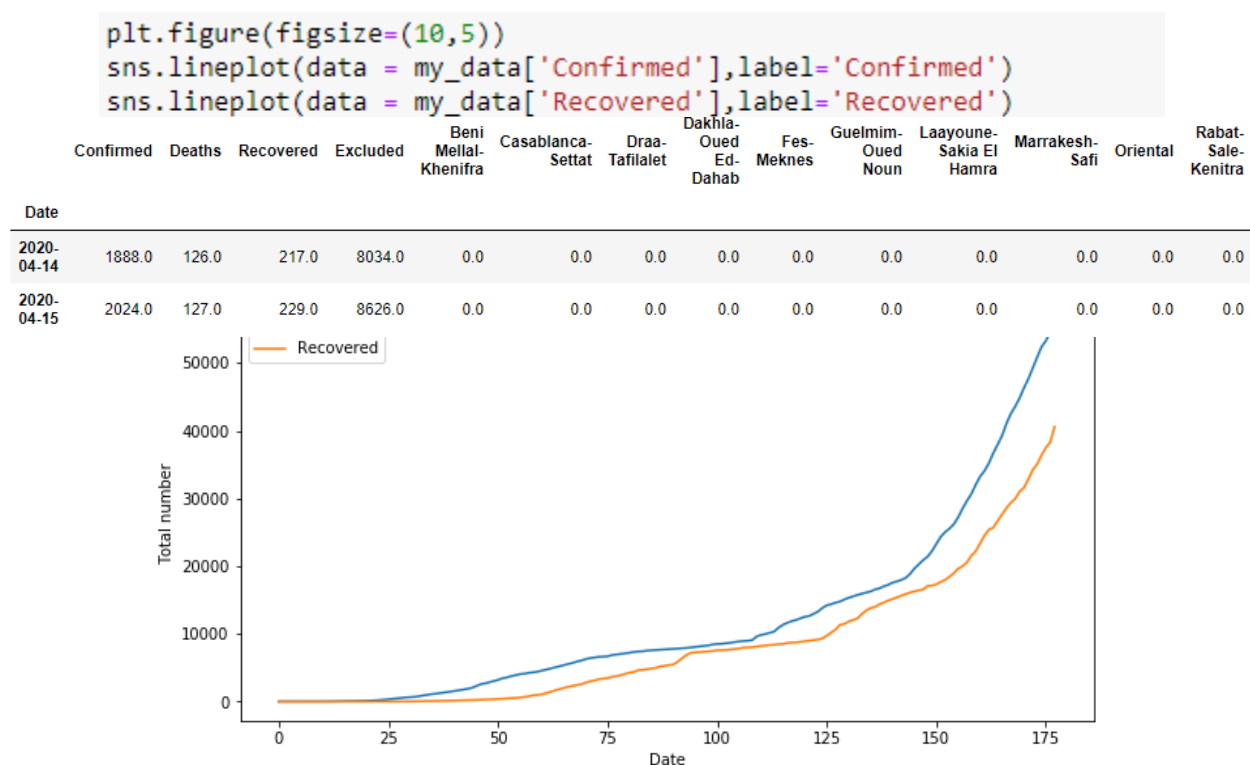
Résultats :

	Confirmed	Deaths	Recovered	Excluded	Beni Mellal-Khenifra	Casablanca-Settat	Draa-Tafilalet	Dakhla-Oued Ed-Dahab	Fes-Meknes	Guelmim-Oued Noun	Laayoune-Sakia El Hamra	Marrakesh-Safi	Oriental	Rabat-Sale-Kenitra
Date														
2020-03-02	1.0	0.0	0.0	28.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2020-03-03	0.0	0.0	0.0	32.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2020-03-04	2.0	0.0	0.0	34.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2020-03-05	0.0	0.0	0.0	40.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2020-03-06	0.0	0.0	0.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

- Ajout de cas actif et de tests totaux par jour dans la base de données :

Recovered	Excluded	Beni Mellal-Khenifra	Casablanca-Settat	Draa-Tafilalet	Dakhla-Oued Ed-Dahab	Fes-Meknes	Guelmim-Oued Noun	Laayoune-Sakia El Hamra	Marrakesh-Safi	Oriental	Rabat-Sale-Kenitra	Souss-Massa	Tanger-Tetouan-Al Hoceima	Actif	Total Test
0.0	28.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	29.0
0.0	32.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	32.0
0.0	34.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	36.0
0.0	40.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	40.0
0.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	50.0

- Représentation graphique du nombre de cas confirmés et de recouvrements :



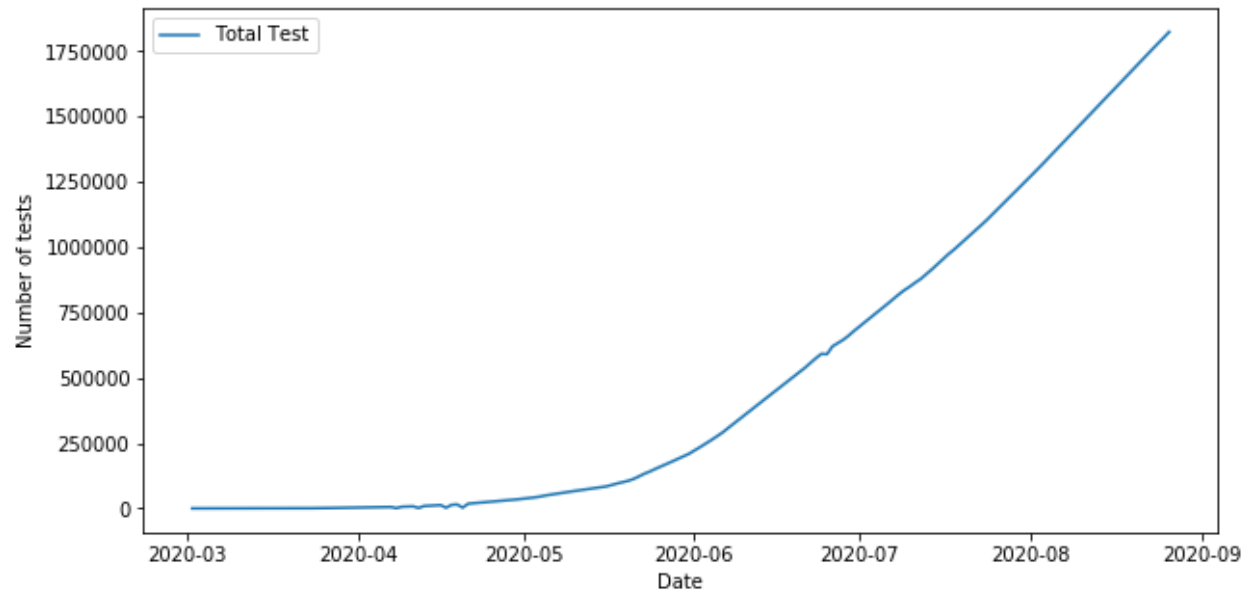
NB : La répartition des nouveaux cas pour les lignes 14 et 15 avril sur les régions n'est pas disponible et cela affectera notre analyse.

La solution optimale pour les données manquantes dans cet exemple est de les supprimer.

```
my_data_ori = my_data.copy()
my_data = my_data[my_data.index != '2020-04-14']
my_data = my_data[my_data.index != '2020-04-15']
```

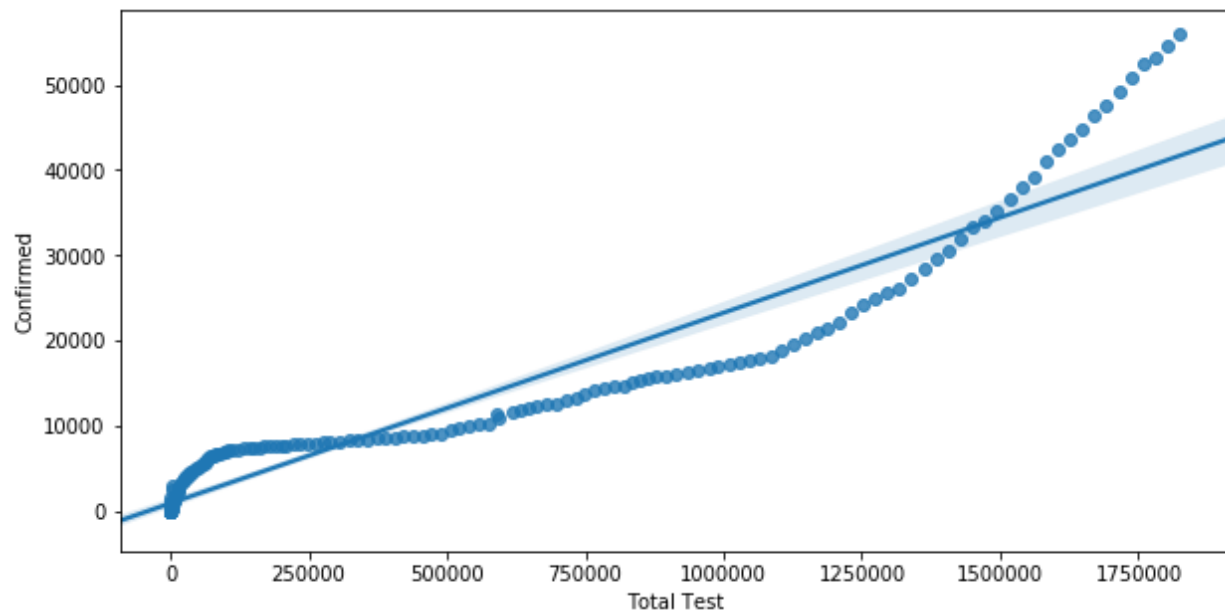
- Augmentations quotidiennes du test total :

```
plt.figure(figsize=(10,5))
sns.lineplot(data = my_data['Total Test'],label='Total Test')
plt.xlabel('Date')
plt.ylabel('Number of tests')
```



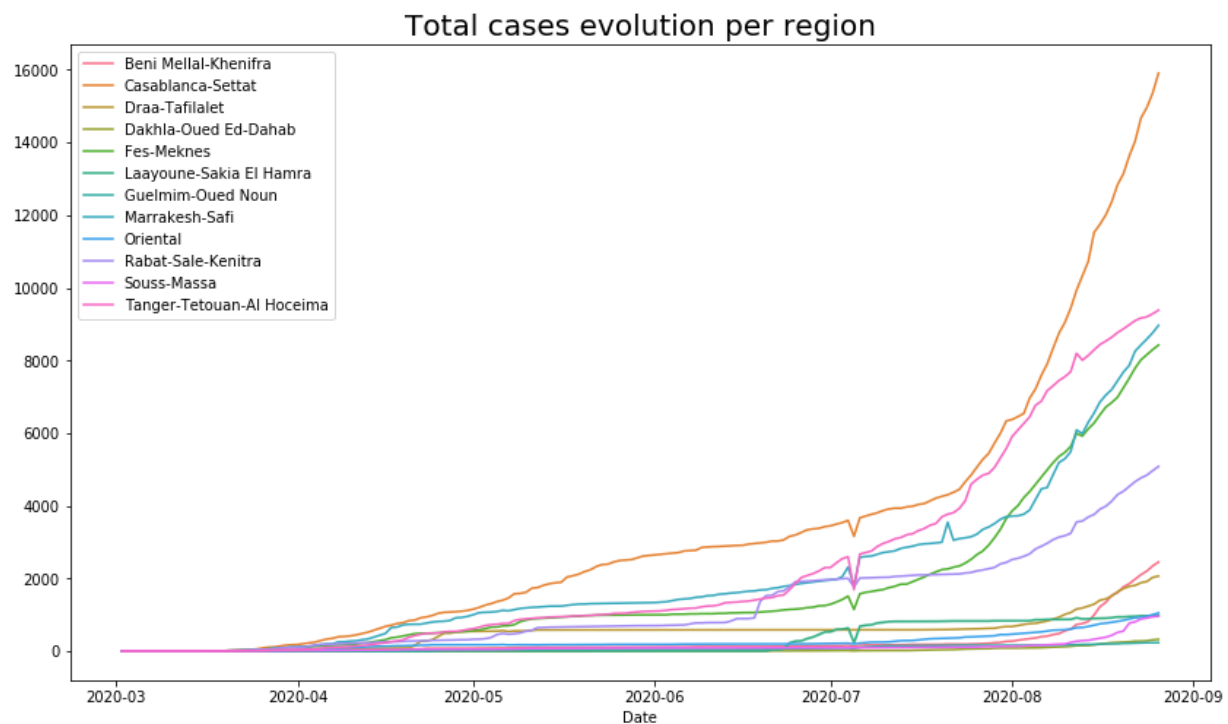
- Corrélation entre le nombre de tests par jour et les nouveaux cas confirmés

```
plt.figure(figsize=(10,5))
sns.regplot(x = my_data['Total Test'],y = my_data['Confirmed'])
```



- Graphique de l'évolution du nombre total de cas par région :

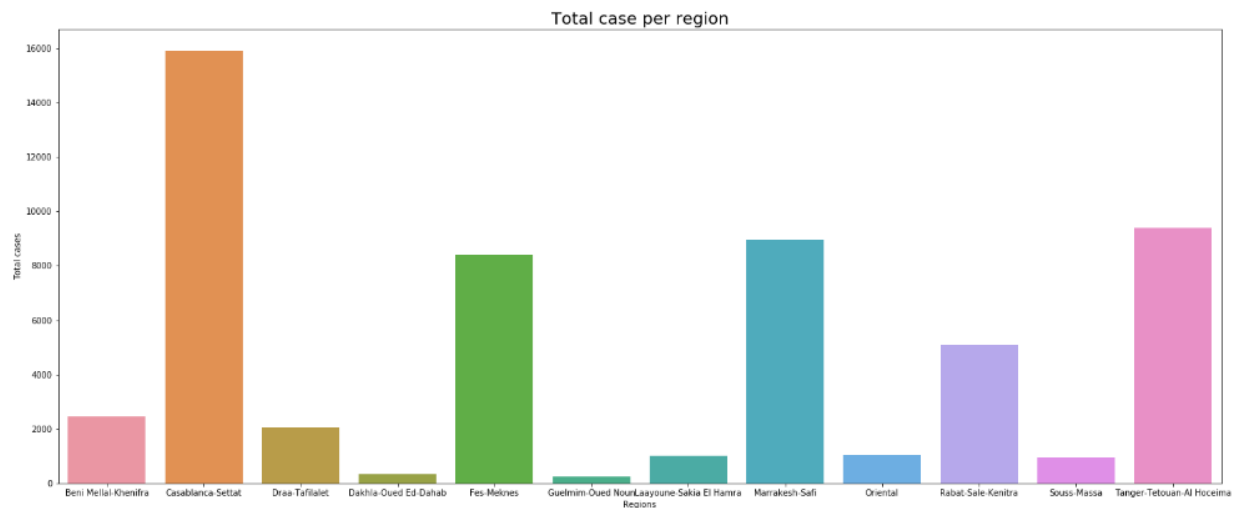
```
Regions1 = ['Beni Mellal-Khenifra',
            'Casablanca-Settat', 'Draa-Tafilalet', 'Dakhla-Oued Ed-Dahab',
            'Fes-Meknes', 'Laayoune-Sakia El Hamra', 'Guelmim-Oued Noun', 'Laayoune-Sakia El Hamra',
            'Marrakesh-Safi', 'Oriental', 'Rabat-Sale-Kenitra', 'Souss-Massa', 'Tanger-Tetouan-Al Hoceima']
plt.figure(figsize=(14,8))
sns.lineplot(data = my_data[Regions1],dashes=False)
plt.title('Total cases evolution per region',fontsize = 20)
```



```

region_bar = my_data.iloc[[-1]].transpose()
region_bar = region_bar.drop(['Confirmed', 'Recovered', 'Deaths', 'Excluded', 'Actif', 'Total Test'])
region_bar.columns=['Total cases']
region_bar.index.name = 'Regions'
plt.figure(figsize=(25,10))
sns.barplot(x=region_bar.index,y='Total cases',data=region_bar)
plt.title('Total case per region',fontsize=20)

```

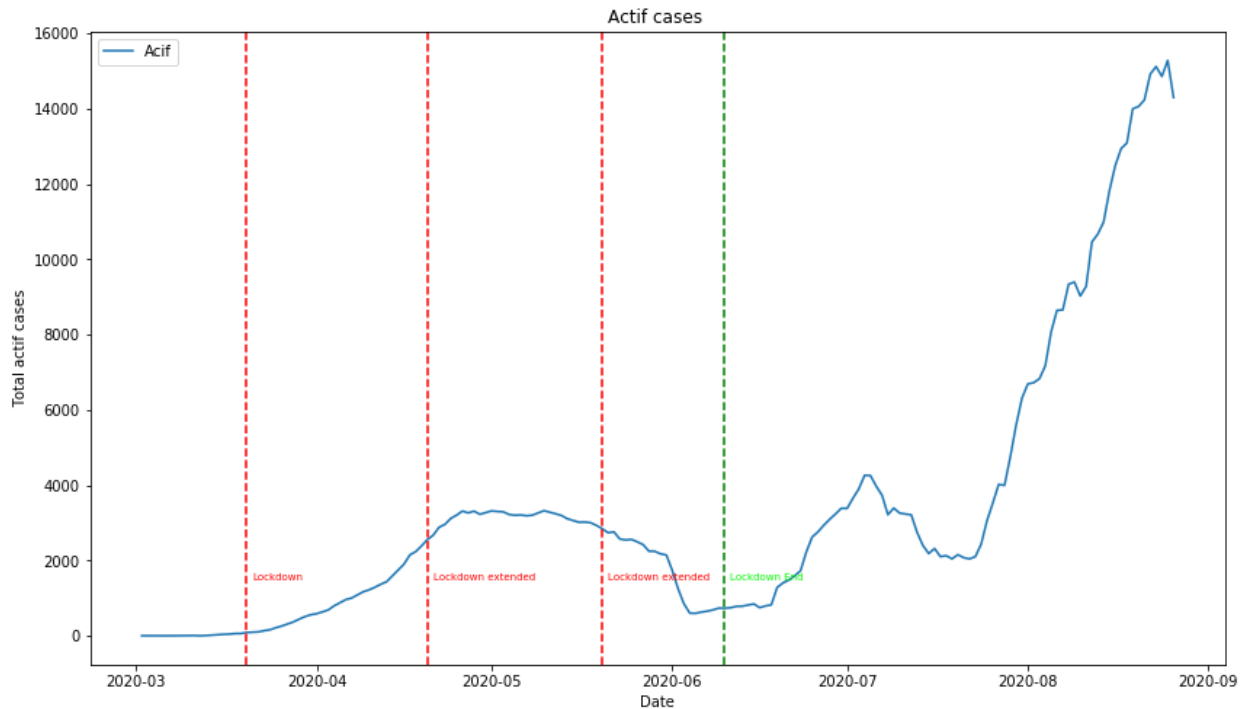


- Représentation graphique des cas actifs

```

plt.figure(figsize=(14,8))
sns.lineplot(data = my_data['Actif'],label='Acif')
plt.axvline('2020-03-20',ls = '--',c = 'r')
plt.axvline('2020-04-20',ls = '--',c = 'r')
plt.axvline('2020-05-20',ls = '--',c = 'r')
plt.axvline('2020-06-10',ls = '--',c = 'g')
plt.text('2020-03-21', 1500, 'Lockdown', fontsize=7,color='#FF0000')
plt.text('2020-04-21', 1500, 'Lockdown extended', fontsize=7,color='#FF0000')
plt.text('2020-05-21', 1500, 'Lockdown extended', fontsize=7,color='#FF0000')
plt.text('2020-06-11', 1500, 'Lockdown End', fontsize=7,color='#00FA00')
plt.xlabel('Date')
plt.ylabel('Total actif cases')
plt.title('Actif cases ')
plt.legend()

```

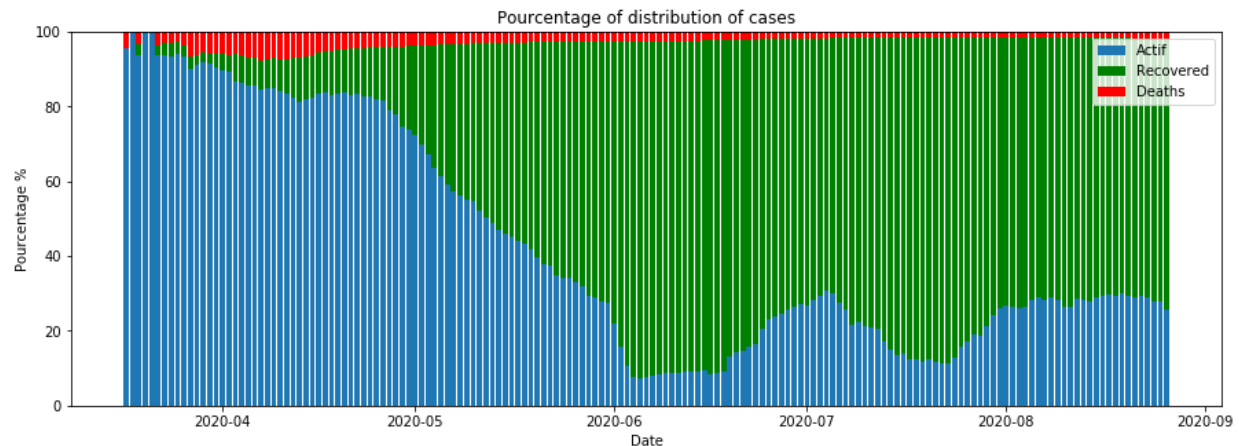


NB : les lignes verticales rouges représentent les périodes de confinement

On remarque une augmentation des cas positif après la fin de confinement.

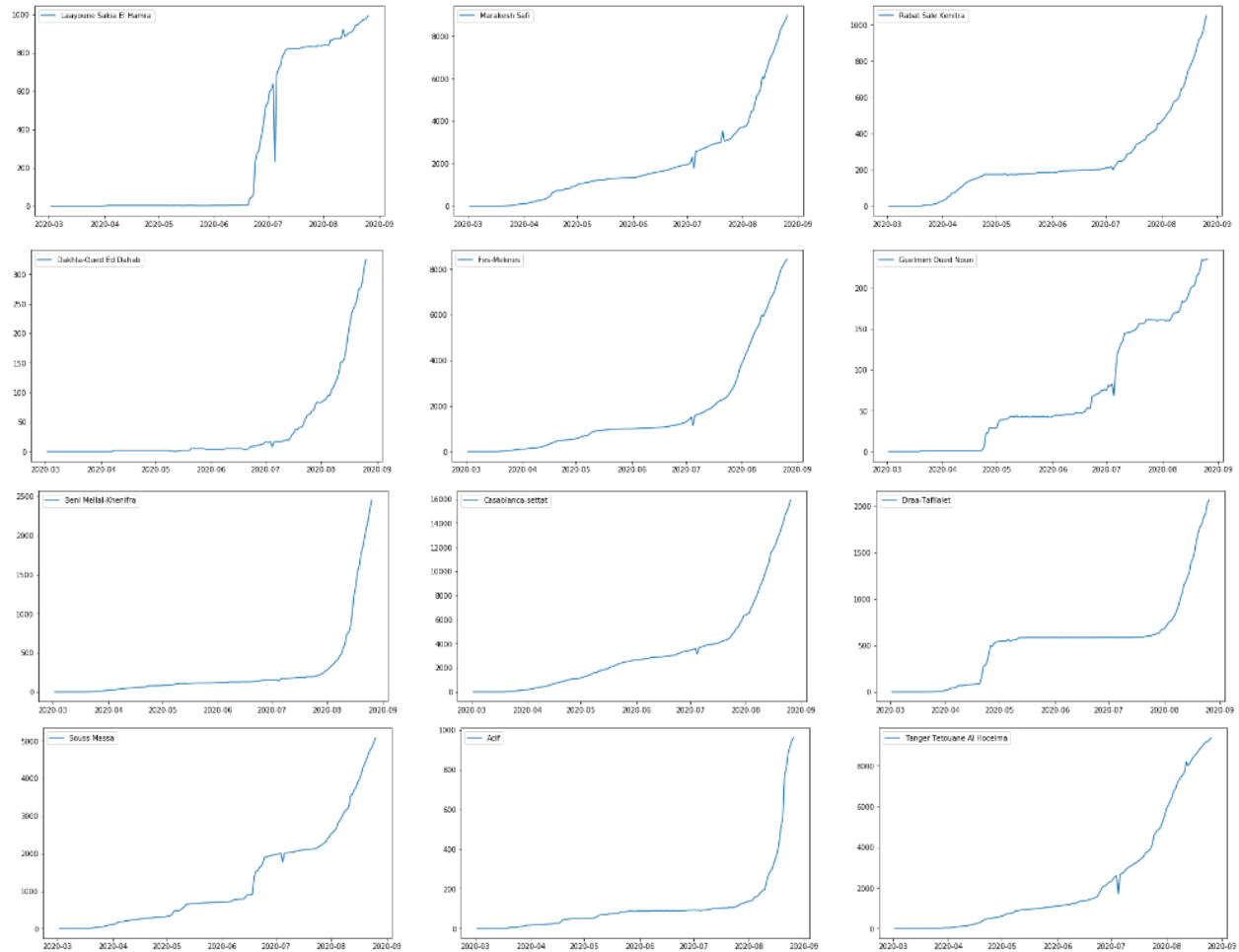
- Pourcentage d'Actif, de décès et de cas recouvrés par jour :

```
plt.figure(figsize=(15,5))
dataVar1 =(my_data_ori['Actif']*100/my_data_ori['Confirmed'])[15:]
dataVar3 =(my_data_ori['Deaths']*100/my_data_ori['Confirmed'])[15:]
dataVar2 =(my_data_ori['Recovered']*100/my_data_ori['Confirmed'])[15:]
p1 = plt.bar(dataVar1.index,
             dataVar1,label='Actif')
p2 = plt.bar(dataVar2.index,
             dataVar2,
             bottom=dataVar1,color='g',label='Recovered')
p3 = plt.bar(dataVar3.index,
             dataVar3,
             bottom=dataVar1+dataVar2,color='r',label='Deaths')
plt.xlabel('Date')
plt.ylabel('Pourcentage %')
plt.title('Pourcentage of distribution of cases')
plt.legend()
```



- Graphique du nombre total de cas par région :

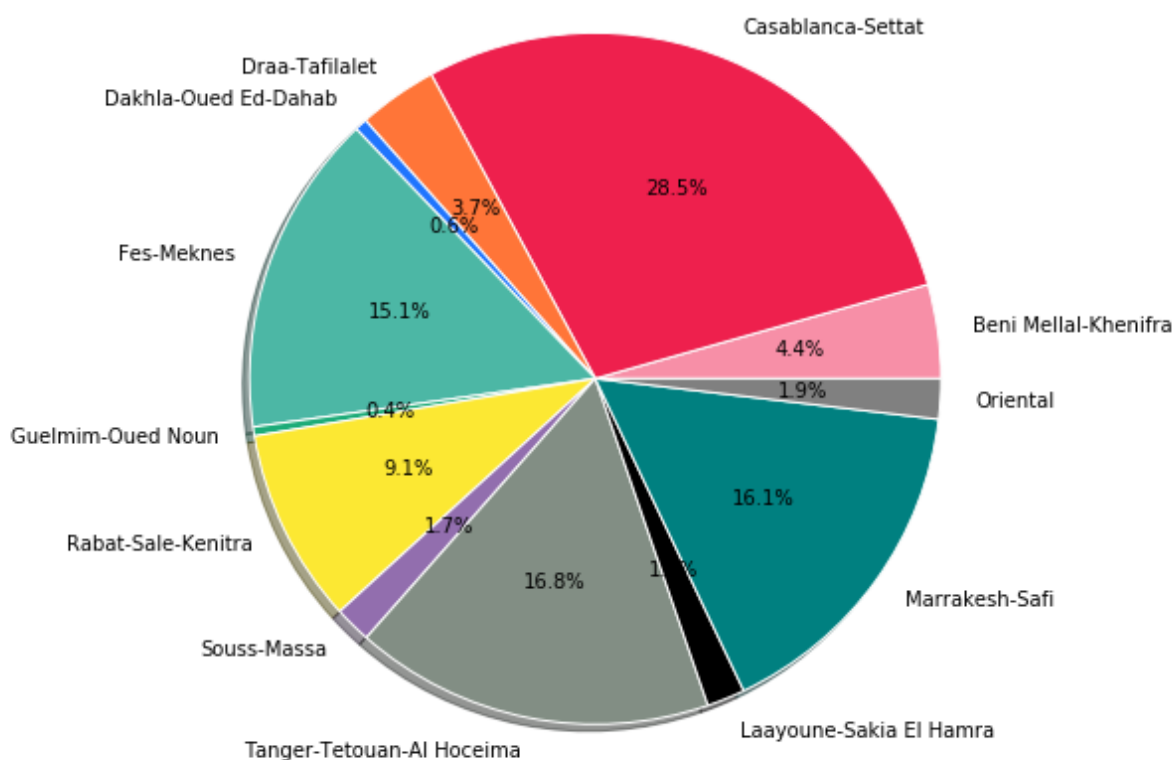
```
plt.figure(figsize=(30,25))
BM = my_data['Beni Mellal-Khenifra']
CS = my_data['Casablanca-Settat']
DT = my_data['Draa-Tafilalet']
DO = my_data['Dakhla-Oued Ed-Dahab']
FM = my_data['Fes-Meknes']
GO = my_data['Guelmim-Oued Noun']
LS = my_data['Laayoune-Sakia El Hamra']
MS = my_data['Marrakesh-Safi']
Or = my_data['Oriental']
RS = my_data['Rabat-Sale-Kenitra']
SM = my_data['Souss-Massa']
TT = my_data['Tanger-Tetouan-Al Hoceima']
ax1 = plt.subplot(4, 3, 1)
sns.lineplot(data = BM,label='Beni Mellal-Khenifra')
ax2 = plt.subplot(4, 3, 2)
sns.lineplot(data = CS,label='Casablanca-settat')
ax3 = plt.subplot(4, 3, 3)
sns.lineplot(data = DT,label='Draa-Tafilalet')
ax4 = plt.subplot(4, 3, 4)
sns.lineplot(data = DO,label='Dakhla-Oued Ed Dahab')
ax5 = plt.subplot(4, 3, 5)
sns.lineplot(data = FM,label='Fes-Meknes')
ax6 = plt.subplot(4, 3, 6)
sns.lineplot(data = GO,label='Guelmim Oued Noun')
ax7 = plt.subplot(4, 3, 7)
sns.lineplot(data = LS,label='Laayoune Sakia El Hamra')
ax8 = plt.subplot(4, 3, 8)
sns.lineplot(data = MS,label='Marakesh Safi')
ax9 = plt.subplot(4, 3, 9)
sns.lineplot(data = Or,label='Rabat Sale Kenitra')
ax10 = plt.subplot(4, 3, 10)
sns.lineplot(data = RS,label='Souss Massa')
ax11 = plt.subplot(4, 3, 11)
sns.lineplot(data = SM,label='Acif')
ax12 = plt.subplot(4, 3, 12)
sns.lineplot(data = TT,label='Tanger Tetouane Al Hoceima')
```



- Graphique circulaire du nombre total de cas par région :

```
plt.figure(figsize=(16,8))
Slice = [my_data['Beni Mellal-Khenifra'][-1],my_data['Casablanca-Settat'][-1],my_data['Draa-Tafilalet'][-1],
my_data['Dakhla-Oued Ed-Dahab'][-1],my_data['Fes-Meknes'][-1],my_data['Guelmim-Oued Noun'][-1],
my_data['Rabat-Sale-Kenitra'][-1],my_data['Souss-Massa'][-1],my_data['Tanger-Tetouan-Al Hoceima'][-1],
my_data['Laayoune-Sakia El Hamra'][-1],my_data['Marrakesh-Safi'][-1],my_data['Oriental'][-1]]
Labels = ['Beni Mellal-Khenifra','Casablanca-Settat','Draa-Tafilalet','Dakhla-Oued Ed-Dahab','Fes-Meknes',
'Guelmim-Oued Noun','Rabat-Sale-Kenitra','Souss-Massa','Tanger-Tetouan-Al Hoceima','Laayoune-Sakia El Hamra',
'Marrakesh-Safi','Oriental']
Colors = ['#f78fa7','#EE204D','#FF7538','#1F75FE','#4CB7A5','#1CAC78','#FCE833', '#926EAE', '#828E84',
'#000000', '#008080','#808080']
Explode = [0,0,0,0,0,0,0,0,0,0,0,0]
plt.pie(Slice,colors = Colors,explode = Explode,shadow = True,autopct='%1.1f%%',
labels = Labels,wedgeprops = {'edgecolor':'white'})
plt.title('Pie Chart of Total cases per region')
```

Pie Chart of Total cases per region



Casablanca-Settat, Tanger-Tétouan Al Hoceima, Marrakech Safi ont les plus grandes valeurs de l'ensemble des cas, en raison de l'épicentre commercial, industriel et familial, en particulier pour les villes de Casablanca et Tanger qui représentent les plus grandes zones industrielles au Maroc.

4. Prévision du nombre total de cas - décès et cas confirmés

Nous utilisons Prophet, une procédure de prévision des données de séries chronologiques basée sur un modèle additif où les tendances non linéaires correspondent à la saisonnalité annuelle, hebdomadaire et quotidienne, plus les effets des vacances. Il fonctionne mieux avec des séries chronologiques qui ont de forts effets saisonniers et plusieurs saisons de données historiques. Prophet est robuste pour les données manquantes et les changements dans la tendance, et gère généralement bien les valeurs aberrantes.

- Ajout d'une nouvelle colonne de cas quotidiens à nos données :

```
new_cases = []
for i in range(len(total_Confirmed)):
    if i == 0:
        new_cases.append(0)
    elif total_Confirmed[i] < total_Confirmed[i-1]:
        new_cases.append(0)
    else:
        temp = int(total_Confirmed[i] - total_Confirmed[i-1])
        new_cases.append(temp)

new_cases = np.array(new_cases)
my_data['New cases'] = new_cases
```

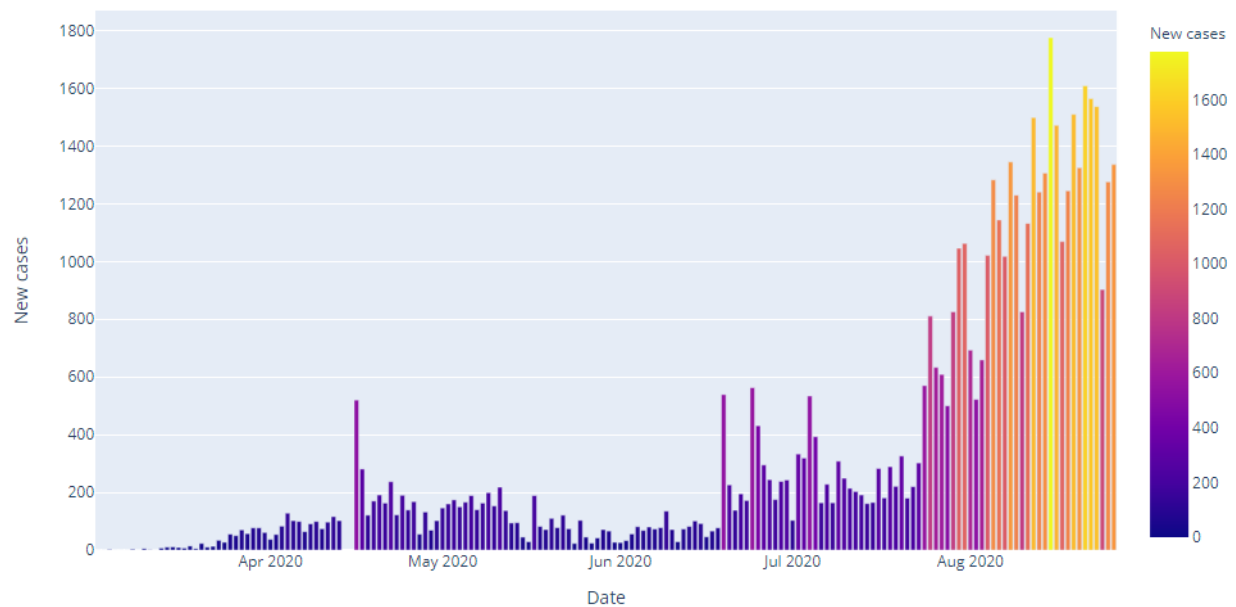
```
df = pd.DataFrame(my_data['New cases'])
df.style.background_gradient(cmap='Reds')
```

New cases		2020-08-15 00:00:00	1776
Date		2020-08-16 00:00:00	1472
2020-03-02 00:00:00	0	2020-08-17 00:00:00	1069
2020-03-03 00:00:00	0	2020-08-18 00:00:00	1245
2020-03-04 00:00:00	2	2020-08-19 00:00:00	1510
2020-03-05 00:00:00	0	2020-08-20 00:00:00	1325
2020-03-06 00:00:00	0	2020-08-21 00:00:00	1609
2020-03-07 00:00:00	0	2020-08-22 00:00:00	1565
2020-03-08 00:00:00	3	2020-08-23 00:00:00	1537
2020-03-09 00:00:00	0	2020-08-24 00:00:00	903
2020-03-10 00:00:00	5	2020-08-25 00:00:00	1276
2020-03-11 00:00:00	1	2020-08-26 00:00:00	1336

- Mains d'intrigue pour visualiser la situation actuelle :

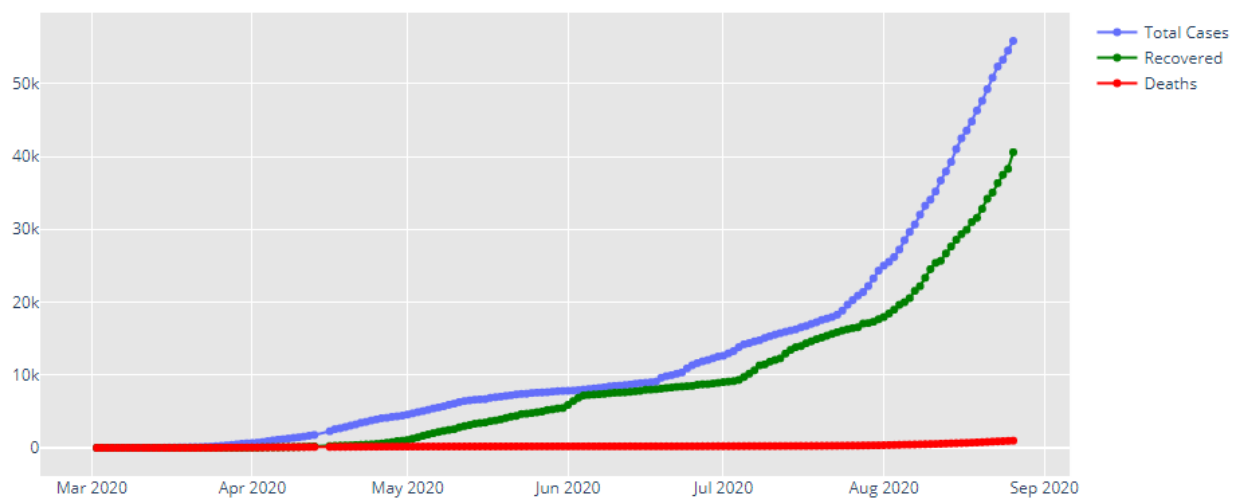
```
fig = px.bar(my_data, x=my_data.index, y="New cases", color='New cases', orientation='v', height=600,
             title='Confirmed Cases in Morocco', color_discrete_sequence = px.colors.cyclical.IceFire)
fig.show()
```

Confirmed Cases in Morocco



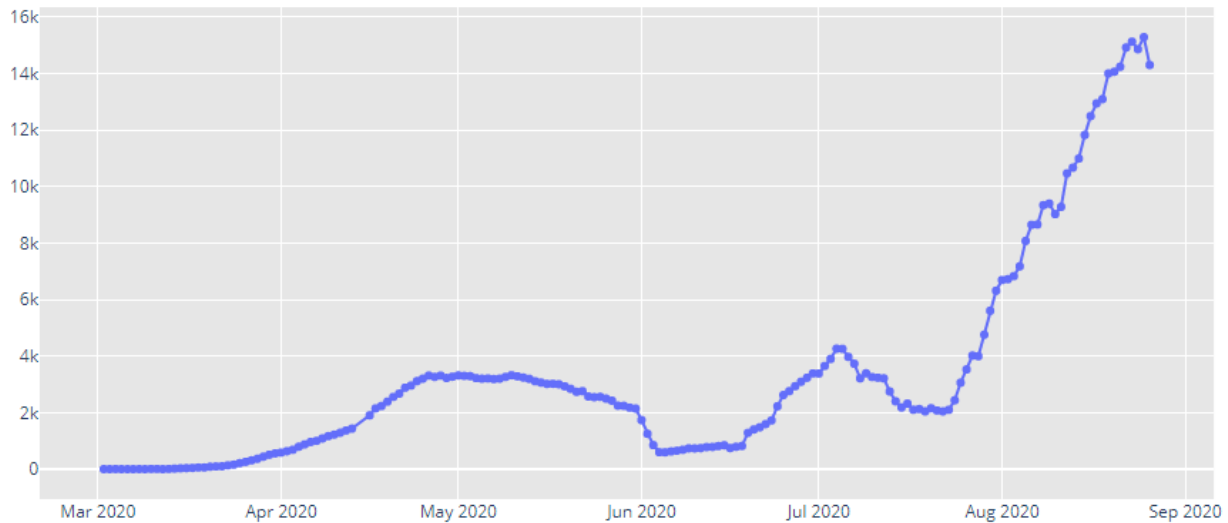
```
fig = go.Figure()
fig.add_trace(go.Scatter(x=my_data.index, y = my_data['Confirmed'], mode= 'lines+markers',name='Total Cases'))
fig.add_trace(go.Scatter(x=my_data.index, y = my_data['Recovered'], mode='lines+markers',name='Recovered',
                        line=dict(color='Green', width=2)))
fig.add_trace(go.Scatter(x=my_data.index, y=my_data['Deaths'], mode='lines+markers', name='Deaths',
                        line=dict(color='Red', width=2)))
fig.update_layout(title_text='Trend of Coronavirus Cases in Morocco (Cumulative cases)',plot_bgcolor='rgb(230, 230, 230)')
fig.show()
```

Trend of Coronavirus Cases in Morocco (Cumulative cases)



```
fig = go.Figure()
fig.add_trace(go.Scatter(x=my_data.index, y = my_data['Actif'], mode= 'lines+markers',name='Total Cases'))
fig.update_layout(title_text='Trend of Actif Coronavirus Cases in Morocco ',plot_bgcolor='rgb(230, 230, 230)')
fig.show()
```

Trend of Actif Coronavirus Cases in Morocco



1.1 Prédéterminer la tendance des cas confirmés dans les 10 prochains jours

L'entrée dans Prophet est toujours une base de données avec deux colonnes : ds et y. La colonne ds (datestamp) doit être d'un format attendu par Pandas, idéalement AAAA-MM-JJ pour une date ou AAAA-MM-JJ HH:MM:SS pour un horodatage. La colonne y doit être numérique et représente la mesure que nous souhaitons prévoir.

```
confirmed = my_data.groupby('Date').sum()['Confirmed'].reset_index()
confirmed.columns = ['ds', 'y']
```

Générer une prévision de 10 jours à l'avance des cas confirmés en utilisant Prophet, avec un intervalle de prédiction de 95 % en créant un modèle de base sans ajustement des paramètres liés à la saisonnalité et des régressions supplémentaires.

```
m = Prophet(interval_width=0.95)
m.fit(confirmed)
future = m.make_future_dataframe(periods=10)
future.tail(10)
```

	ds
176	2020-08-27
177	2020-08-28
178	2020-08-29
179	2020-08-30
180	2020-08-31
181	2020-09-01
182	2020-09-02
183	2020-09-03
184	2020-09-04
185	2020-09-05

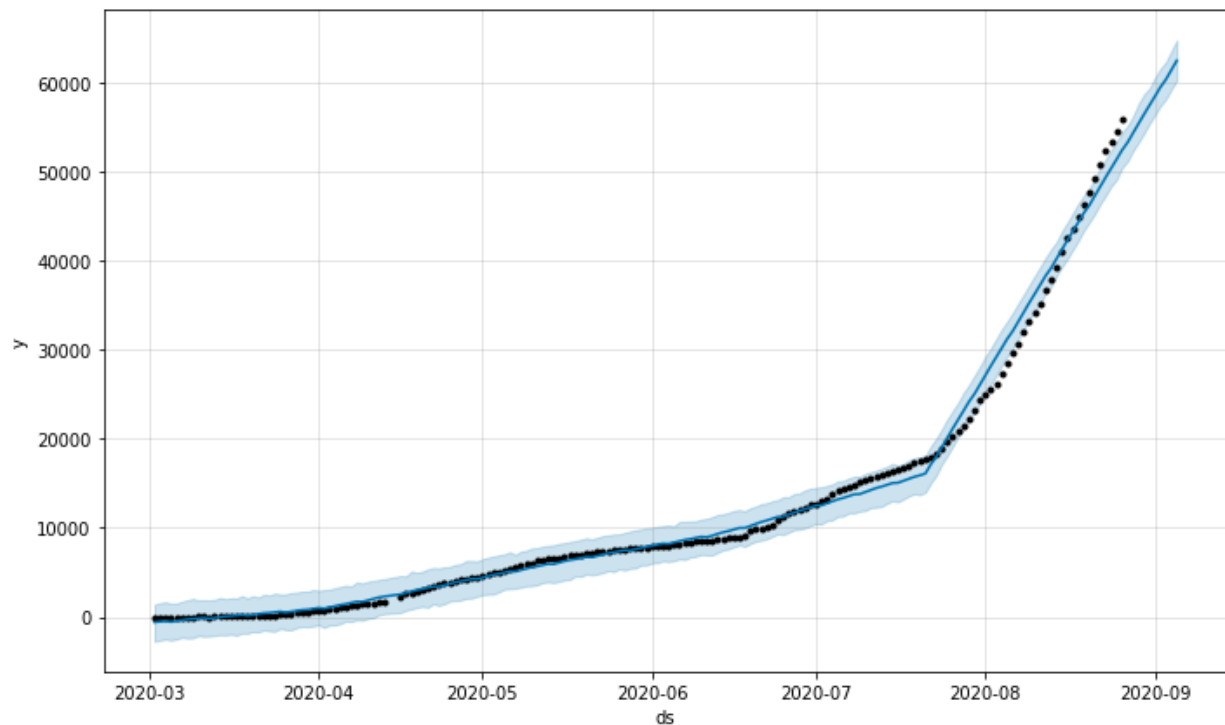
- Prédire l'avenir avec la date, et les limites supérieure et inférieure de la valeur y :

```
forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

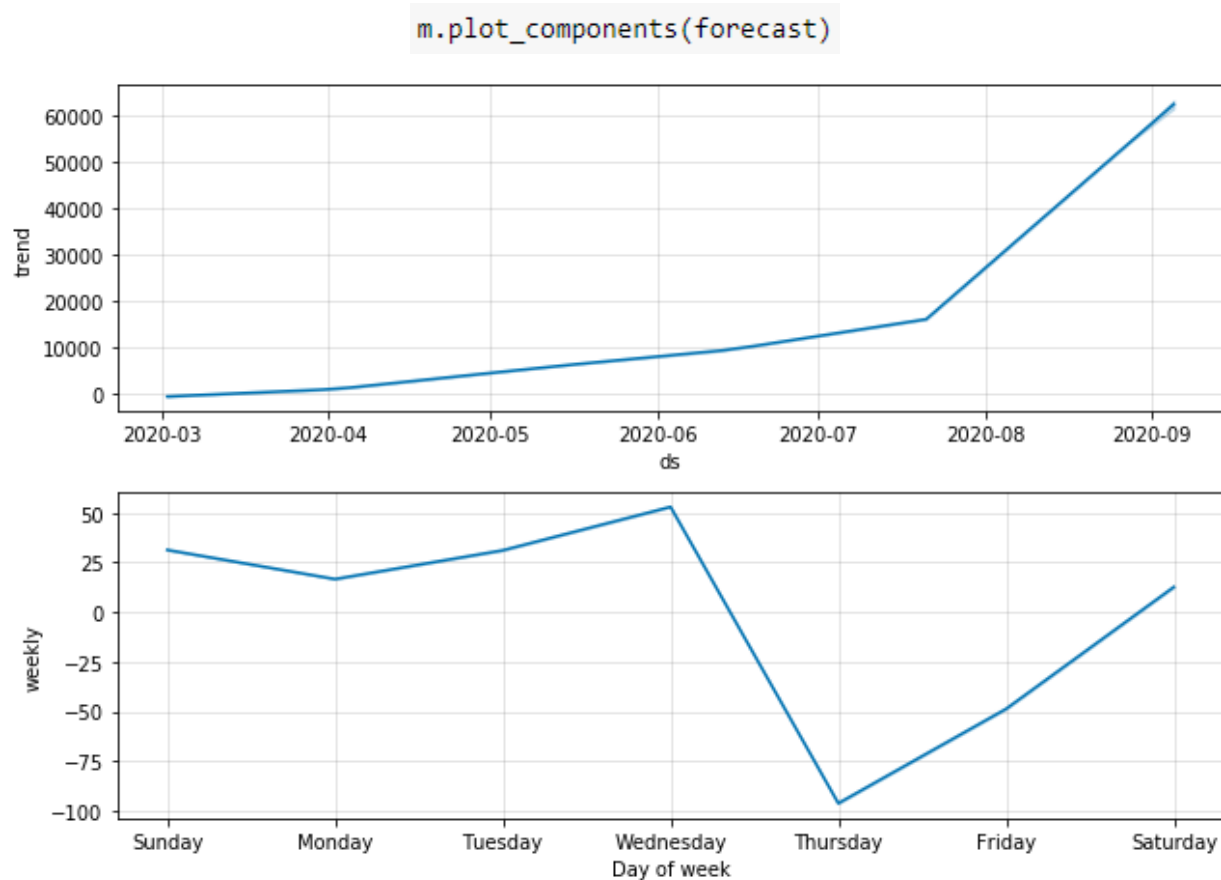
	ds	yhat	yhat_lower	yhat_upper
181	2020-09-01	58413.141760	56337.730700	60551.895219
182	2020-09-02	59442.832555	57347.009653	61543.560868
183	2020-09-03	60301.529922	58115.021262	62285.031504
184	2020-09-04	61356.871219	59200.760383	63533.678581
185	2020-09-05	62426.017927	60207.604430	64719.370829

- Tracer les prévisions futures :

```
m.plot(forecast)
```

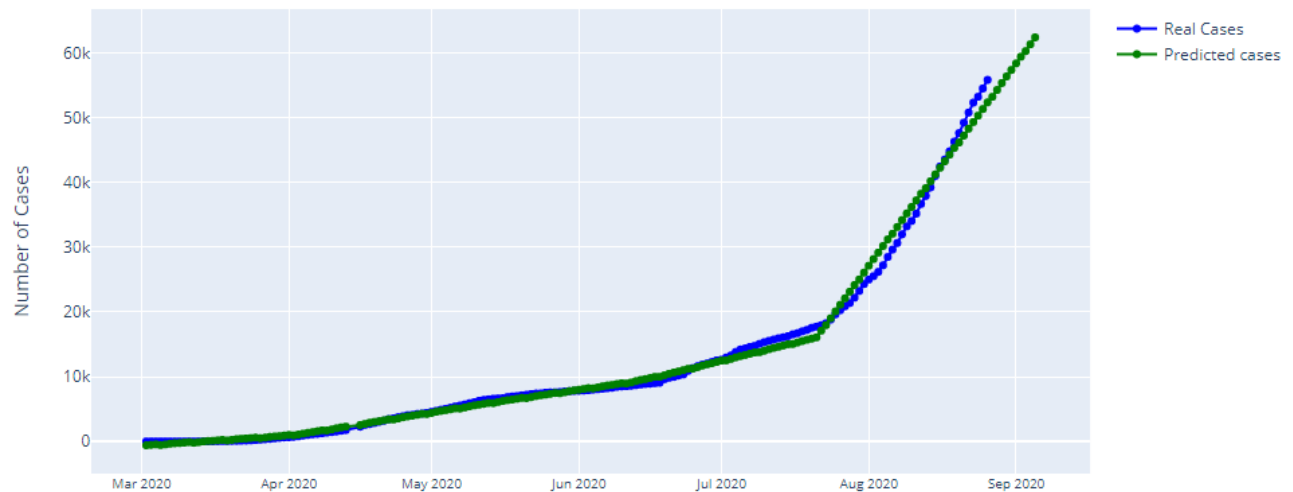


- La distribution hebdomadaire :



- Les cas réels à partir de notre base de données par rapport aux cas prédits par le modèle :

```
plt.figure(figsize=(20,12))
fig = go.Figure()
fig.add_trace(go.Scatter(x=confirmed['ds'], y=confirmed['y'], mode='lines+markers', name='Real Cases',
                        line=dict(color='blue', width=2)))
fig.add_trace(go.Scatter(x=forecast['ds'], y=forecast['yhat'], mode='lines+markers', name='Predicted cases',
                        line=dict(color='green', width=2)))
fig.update_layout(title='Real Covid Cases vs Predicted Cases In Morocco ', xaxis_tickfont_size=10 ,
                  yaxis=dict(title='Number of Cases'))
```



Le modèle prédit plus de nouveaux cas au Maroc.

- Testez notre modèle :

```
df_cv = cross_validation(m, initial='60 days', period='10 days', horizon = '10 days')
df_cv
```

	ds	yhat	yhat_lower	yhat_upper	y	cutoff
0	2020-05-09	5768.529324	5702.050927	5832.747359	5910.0	2020-05-08
1	2020-05-10	5919.193509	5850.524253	5988.454289	6063.0	2020-05-08
2	2020-05-11	6048.972193	5974.563669	6127.387614	6281.0	2020-05-08
3	2020-05-12	6191.281794	6098.635556	6288.619554	6418.0	2020-05-08
4	2020-05-13	6330.445328	6216.766696	6452.590227	6512.0	2020-05-08
...
105	2020-08-22	40865.730927	38396.905721	43341.886330	50812.0	2020-08-16
106	2020-08-23	41557.957471	39056.607724	43955.520571	52349.0	2020-08-16
107	2020-08-24	42000.859541	39733.953350	44339.118770	53252.0	2020-08-16
108	2020-08-25	42656.796942	40303.544127	44886.970346	54528.0	2020-08-16
109	2020-08-26	43345.518432	41073.965930	45739.275324	55864.0	2020-08-16

1.2 Prédire les nouveaux cas quotidiens

```
new_cases = my_data.groupby('Date').sum()['New cases'].reset_index()
new_cases.columns = ['ds', 'y']
```

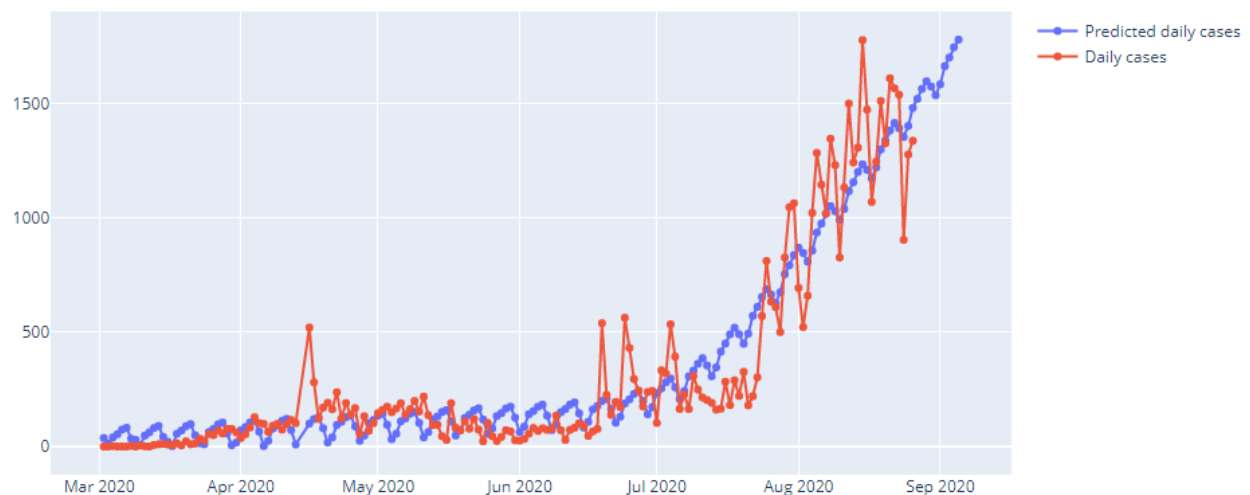
```
m_1 = Prophet(interval_width=0.95)
m_1.fit(new_cases)
future_1 = m_1.make_future_dataframe(periods=10)
future_1.tail()
```

	ds
181	2020-09-01
182	2020-09-02
183	2020-09-03
184	2020-09-04
185	2020-09-05

```
forecast_1 = m_1.predict(future_1)
forecast_1[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

	ds	yhat	yhat_lower	yhat_upper
181	2020-09-01	1582.982752	1303.989110	1837.593519
182	2020-09-02	1661.676506	1391.796426	1935.872000
183	2020-09-03	1700.436432	1423.822942	1965.312502
184	2020-09-04	1744.747038	1471.281357	2007.932006
185	2020-09-05	1777.961953	1509.954985	2053.129589

```
fig = go.Figure()
fig.add_trace(go.Scatter(x=forecast_1.ds, y = abs(forecast_1['yhat']).round(), mode= 'lines+markers',
                        name='Predicted daily cases'))
fig.add_trace(go.Scatter(x=my_data.index, y = my_data['New cases'], mode= 'lines+markers',name='Daily cases'))
```



Comme le montre le modèle, les cas quotidiens sont croissant pour les 10 prochains jours.

1.3 Prédire le nombre total de décès

```
deaths = my_data.groupby('Date').sum()['Deaths'].reset_index()
deaths.columns = ['ds', 'y']
```

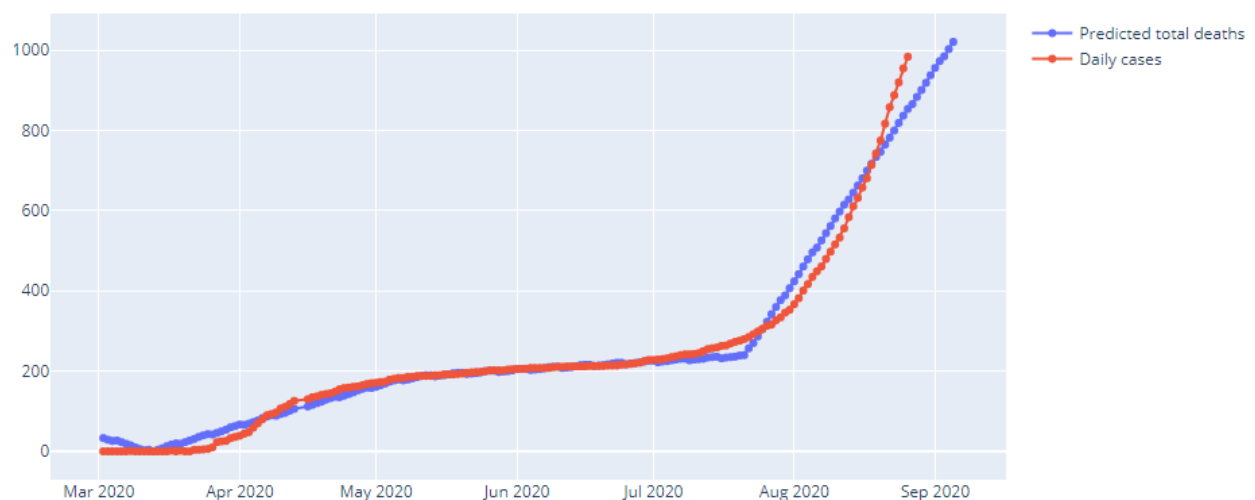
```
m_2 = Prophet(interval_width=0.95)
m_2.fit(deaths)
future_2 = m_2.make_future_dataframe(periods=10)
future_2.tail()
```

	ds
181	2020-09-01
182	2020-09-02
183	2020-09-03
184	2020-09-04
185	2020-09-05

```
forecast_2 = m_2.predict(future_2)
forecast_2[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

	ds	yhat	yhat_lower	yhat_upper
181	2020-09-01	955.963303	901.438406	1019.774494
182	2020-09-02	972.835771	919.522027	1032.061152
183	2020-09-03	985.478555	928.779214	1045.945507
184	2020-09-04	1003.067466	942.693583	1066.009478
185	2020-09-05	1020.709849	960.052675	1081.512267

```
fig = go.Figure()
fig.add_trace(go.Scatter(x=forecast_2.ds, y = abs(forecast_2['yhat']).round(), mode= 'lines+markers',
                        name='Predicted total deaths'))
fig.add_trace(go.Scatter(x=my_data.index, y = my_data['Deaths'], mode= 'lines+markers', name='Daily cases'))
```



CONCLUSION

Ce travail permet d'offrir des aperçus, des analyses, et des prédictions en utilisant les données offertes par le ministère de la Santé pour aider à prendre des décisions futures par les autorités pour empêcher notre peuple des pires situations à venir.

Les résultats de ce travail peuvent être améliorés en se basant sur une base de données qui est pertinente et qui représente les données réelles en toute fidélité.