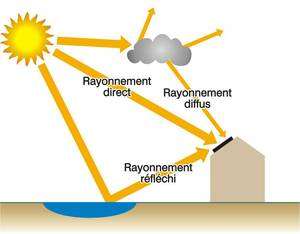
Projet FDEC 2018

Sujet : Défi EGC 2018

# Présentation de la problématique

L’ile de la Réunion est en train d’installer des panneaux solaires afin d’obtenir une autonomie énergétique. Nous avons donc des données sur l’ensoleillement, celle-ci ont été mesure par des capteur SPN1 (Sunshine Pyranometer). Le rayonnement solaire peut être décomposer en 3 flux :

* Le flux global FGlobal
* Le flux diffus FDiffus
* Le flux direct FDirect: FDirect = FGlobal - FDiffus



Les panneaux solaires produisent plus d’énergie lorsque le rayonnement direct est fort. Il pourrait donc être intéressant de prédire FDirect ou l’indice de fraction direct (Kb = FDirect / FGlobal). Cela afin de prédire la production d’énergie solaire et adapter la production d’énergie par d’autres méthode en fonction.

15 stations ont été équipé avec les capteurs SPN1 mais également des capteurs météorologiques nous avons donc 7 mesures par stations sur les 5 dernières années :

* FG\_avg (en W=m2) : le flux global
* FD\_avg (en W=m2) : le flux diffus
* Patm\_avg (en hPa) : la pression atmosphérique
* RH\_avg (en %) : le taux d’humidité dans l’air
* Text\_avg (en °C) : la température extérieure
* WD\_MeanUnitVector (en degré) : la direction du vent
* WS\_avg (en m/s) : la vitesse du vent

Les mesures sont effectuées toutes les minutes, nous avons donc un cas de prédiction de séries temporelles.

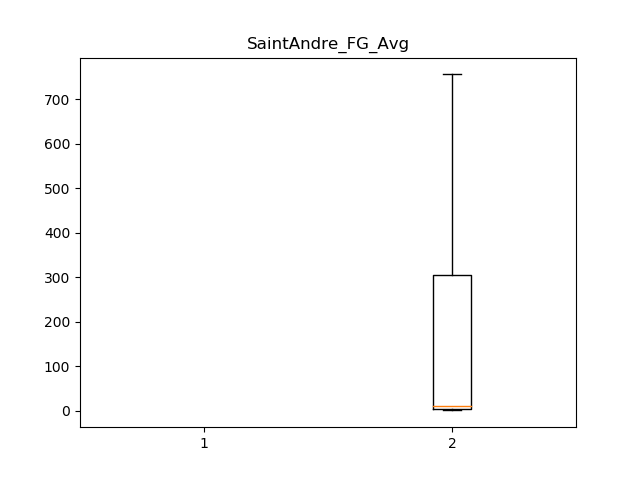
Pour le défi ouvert seulement 2 années (2014-2015) de données de 5 stations étaient disponibles. Il faudra dans un premier temps pré traiter les données, enlever les valeurs aberrantes et gérer les valeurs manquantes. Il sera ensuite possible de dériver les attributs existant en nouveaux (ex : Kb).

Une fois ce nettoyage effectué diverse analyse graphiques pourront être effectuées afin de trouver les attributs les plus intéressants. Celles-ci peuvent également permettre de détecter des problèmes de qualité des données nous amenant à rajouter une étape au pré traitement.

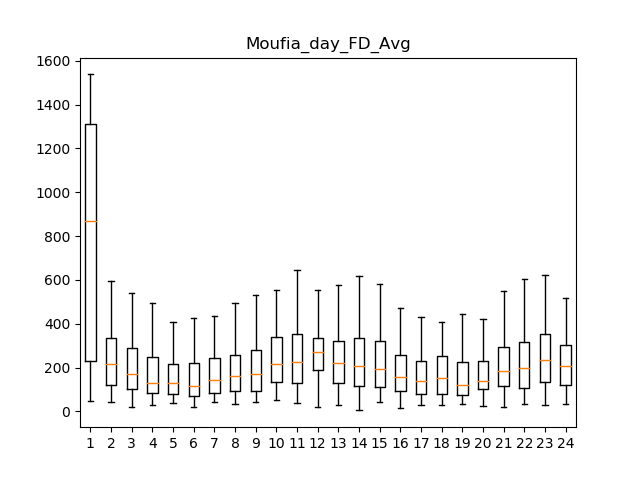
La méthode la plus simple pour effectuer des prédictions sur des série temporelle est la méthode ARIMA.

# Pre traitement

A l’aide d’un rapide affichage graphique on observe que la colonne FG\_Avg de la station Saint André a des valeurs manquantes. Les lignes ou il manque une valeur sont supprimées des dataset.



On observe sur un autre graph qu’il y a des valeurs aberrantes dans les dataset Moufia et Saint André au mois de Janvier 2014 en regardant les csv on voit qu’elles s’arrêtent à la 539e ligne. On enlève donc ces lignes des dataset concernés. Une fois les données nettoyées il est possible de calculer Kb.

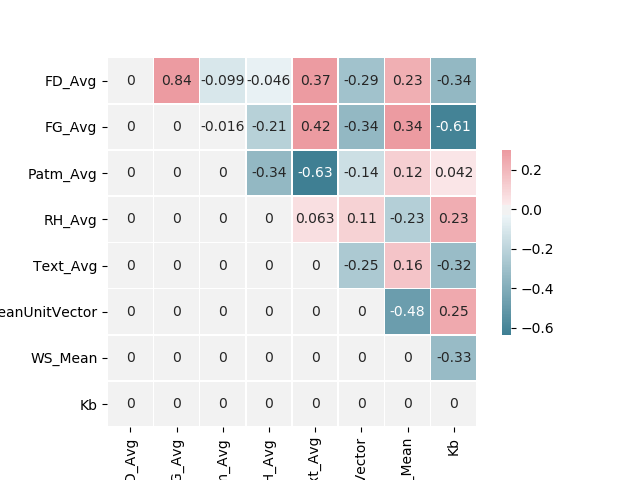
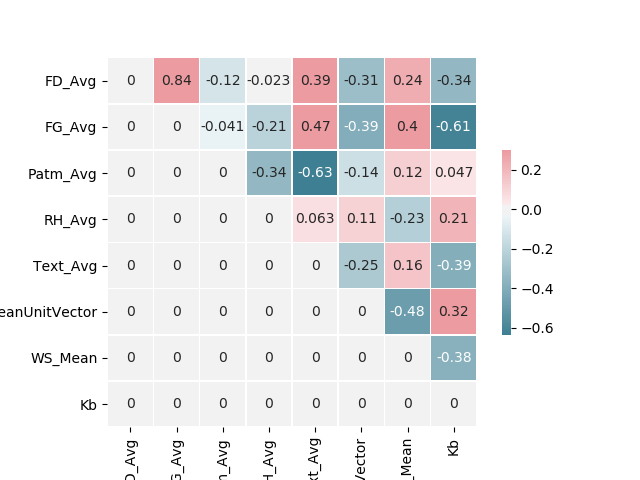


Nous avons encore les données reparties par tranche de 1 minute, il n’est pas forcement intéressant de prédire à la minute près, un découpage par heure serait plus intéressant. Il faut donc faire la moyenne des attributs par tranche d’une heure. Un des avantages de cela est que cette moyenne a aussi un effet anti bruit sur les données.

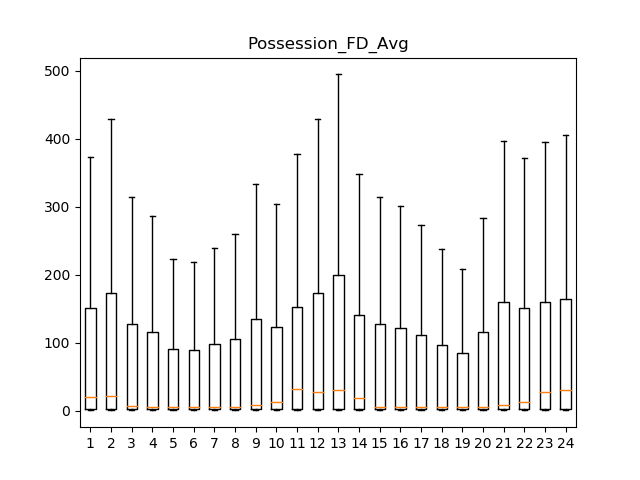
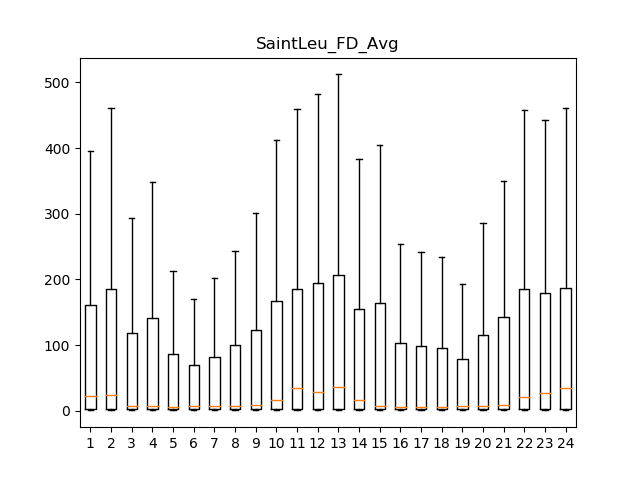
Il peut être également intéressant de supprimer du dataset les données nocturnes, elle risque de perturber notre model alors qu’il est inutile de prédire l’exposition des panneaux lorsqu’on sait qu’il n’y a pas de lumière. Une façon naïve de le faire est de supprimer toutes les lignes comprise entre 20h et 6h du matin (il semblerai que 16h et 8h fonctionnent également).

# Analyse graphique

Pour chaque dataset j’ai trace un graph de corrélation pour chaque couple d’attributs. Le but de ce graph était de trouver les attributs les plus lies avec celui à prédire. Ceux-ci seraient ceux qui apporte le plus à la prédiction. Mais l’on peut voir qu’aucun n’a un fort taux de corrélation avec les attributs que l’on pourrait chercher à trouver (FD\_Avg, Kb). De même si l’on décale FD\_Avg, FG\_Avg et Kb d’une heure (« prédiction ») on n’observe pas non plus de forte corrélation.



Sur les graphs boite a moustache on voit que les valeurs sont proches d’une année à l’autre mais également d’une station a l’autre. Un model qui fonctionne pour l’une des station devrait fonctionner pour les autres mais également ne devrait pas avoir a être modifier au cours du temps.

# Modélisation et prédiction

## Présentation de la méthode ARIMA

La méthode ARIMA est une combinaison de plusieurs méthodes d’analyse de série temporelle, a savoir AR, I et MA.

* Auto Régressif : Il est une représentation d’un processus aléatoire, il fonctionne donc bien pour décrire des évènements naturels. Dans le model AR les valeurs futures du processus dépendent de ses valeurs antérieures.
* Integration : On parle ici de l’ordre de différenciation, la différence entre les valeurs de la série est utilisée. Une série d’ordre 1 donnera yt – yt-1 = µ + εt, une série d’ordre 2 donnera yt – yt-1 – (yt-1 – yt-2) = µ + εt ou µ est une constante. Le but ici est de pouvoir transformer la série afin de la rendre stationnaire, cad que sa moyenne et variance ne change pas dans le temps.
* Moving Average : Similairement au model AR les valeurs futures dépendent des valeurs passe mais ici c’est la distance a la moyenne qui est prise en compte. Pour cette raison MA fonctionne mieux avec une série stationnaire.

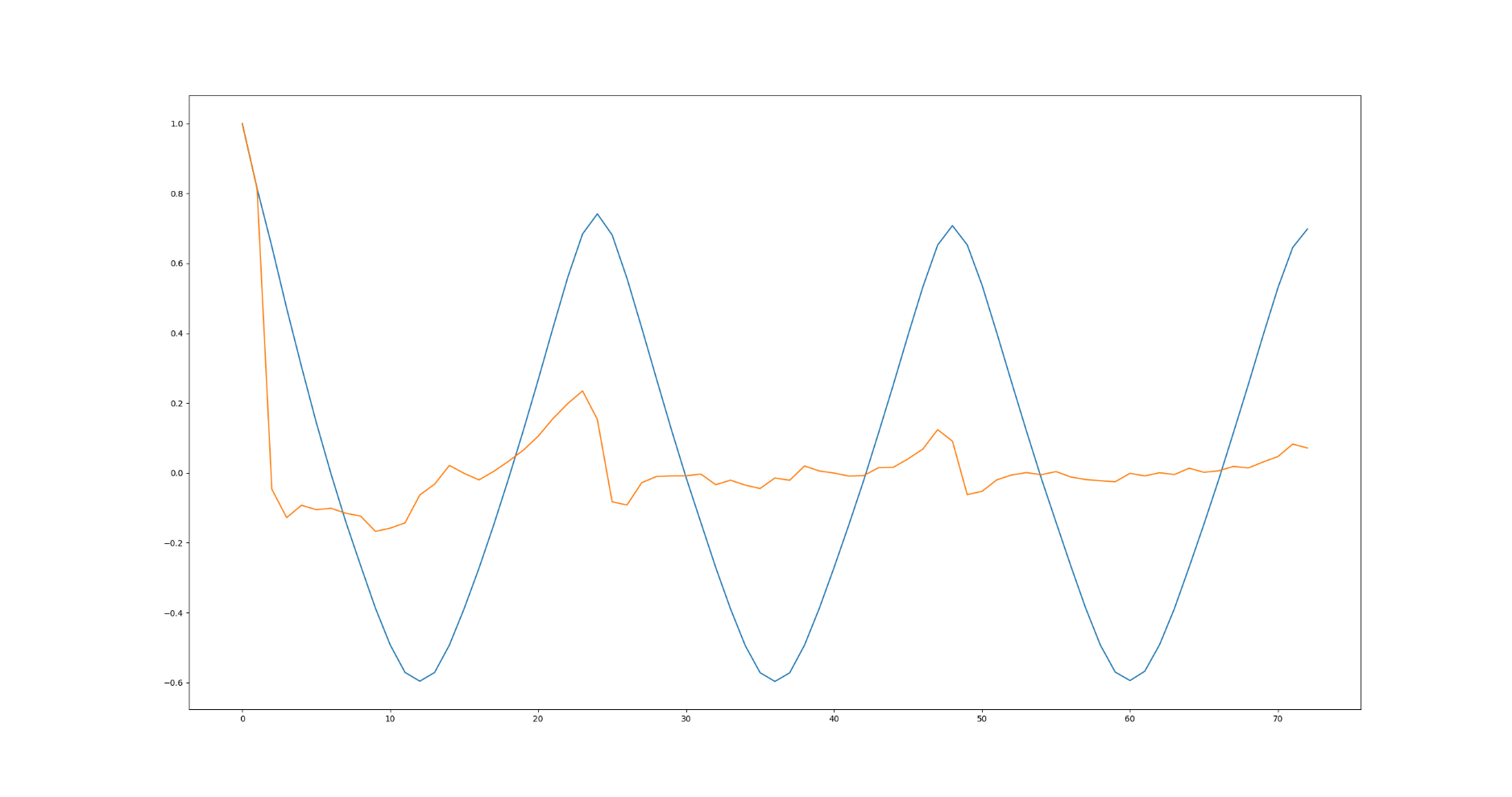
En combinant les différentes parties on obtient un model ARIMA. Celui-ci sera décrit par ses ordres p, d et q. Ou chaque ordre défini l’ordre d’une des partie, ainsi un modèle d’ordre (1, 0, 0) sera équivalent à un model AR(1), etc.

## Mise en place du model

J’ai utilisé python pour expérimenter avec le model ARIMA. Nous allons l’appliquer à un des attributs que nous voulons prédire. Dans un premier temps il est nécessaire de trouver les ordres p, d et q.

### Choix des ordres

Pour choisir l’ordre p une bonne méthode est d’afficher sur un graph l’autocorrélation ainsi que l’autocorrélation partielle de la variable à prédire. Pour Kb de Moufia j’obtiens le graph suivant.



En bleu l’autocorrélation et en orange l’autocorrélation partielle.

On voit sur ce graphique l’autocorrélation est supérieur a 0.5 (0.6 dans ce cas) pour un retard de 2, on pourrait penser que 2 serais une bonne valeur pour p mais si l’on regarde la courbe orange on voit que celle-ci est proche de 0 pour le même retard, cela signifie que la valeur élevée est en fait un effet de celle en 1. On choisit donc p = 1.

La chute rapide de la courbe d’autocorrélation partiel nous indique également que cette série sera mieux expliquée par l’ajout de termes AR que de terme MA, on choisit donc q = 0.

On observe une périodicité (journalière) au niveau de l’autocorrélation on pourrait se demander s’il est nécessaire d’utiliser un model ARIMA saisonnier. Dans le graph précédant on voit que l’autocorrélation partielle reste faible malgré la remontée de l’autocorrélation, il semblerai qu’il ne soit pas nécessaire de tenir compte de la périodicité.

Pour savoir si l’indice d doit être non nul on peut effectuer un test de Dickey-Fuller Augmenté si celui-ci retourne une p-value inferieur a 0.05 notre série est stationnaire. Sur les 5 dataset le p-value est bien inferieur a 0.05 (de l’ordre de 10-17). On en déduit que d=0.

Nous pouvons donc configurer notre model avec (1, 0, 0) il sera donc équivalent à un model AR(1).

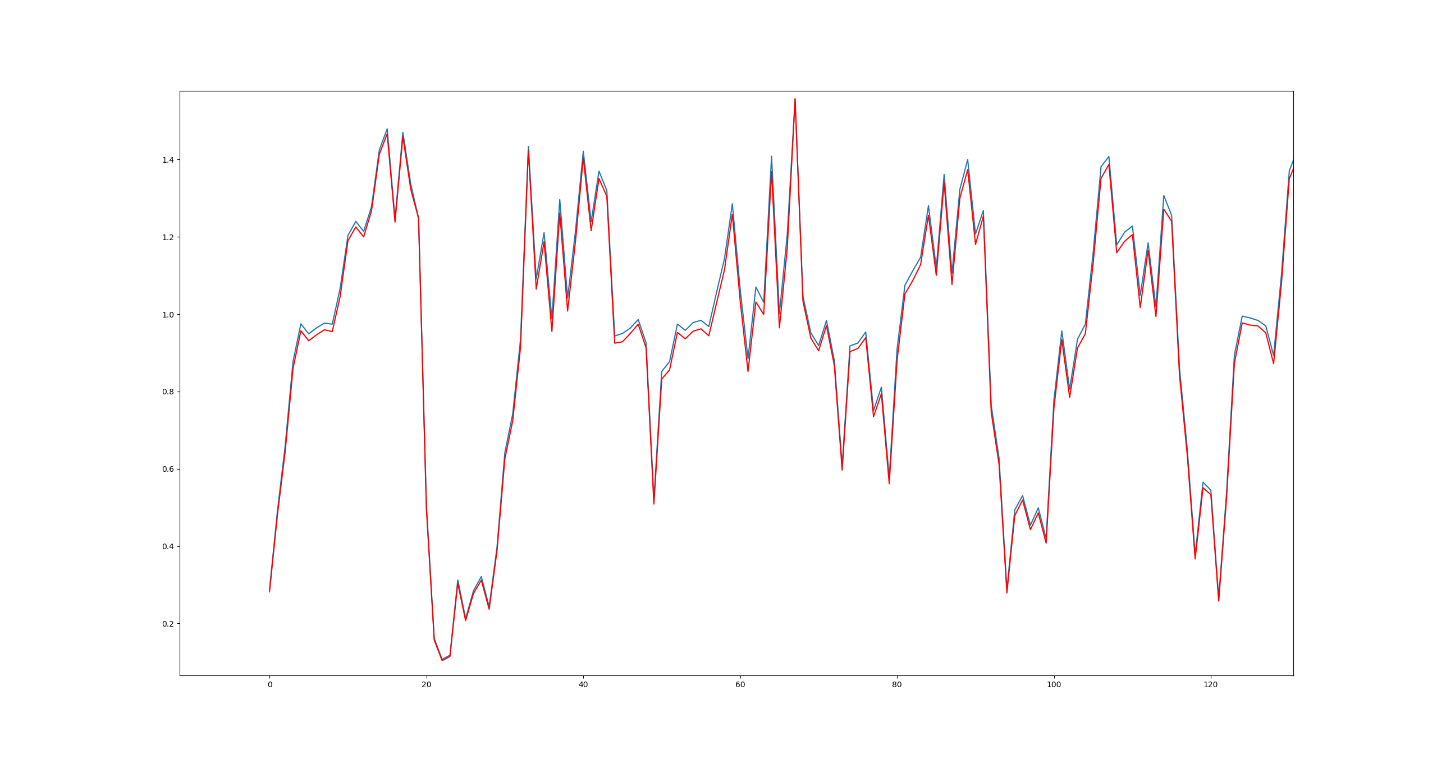
### Evaluation du model

Pour évaluer un model sur une série temporelle on ne peut pas utiliser une validation croise car celle-ci ne garde pas l’ordre des évènements. Mais il est possible d’applique le même principe de séparation entrainement et test. Une façon de faire est de choisir le nombre *n* de valeurs qui vont servir à entrainer le model (par exemple 48, soit 2 jours) puis de comparer la prédiction de ce model avec la valeur observée.

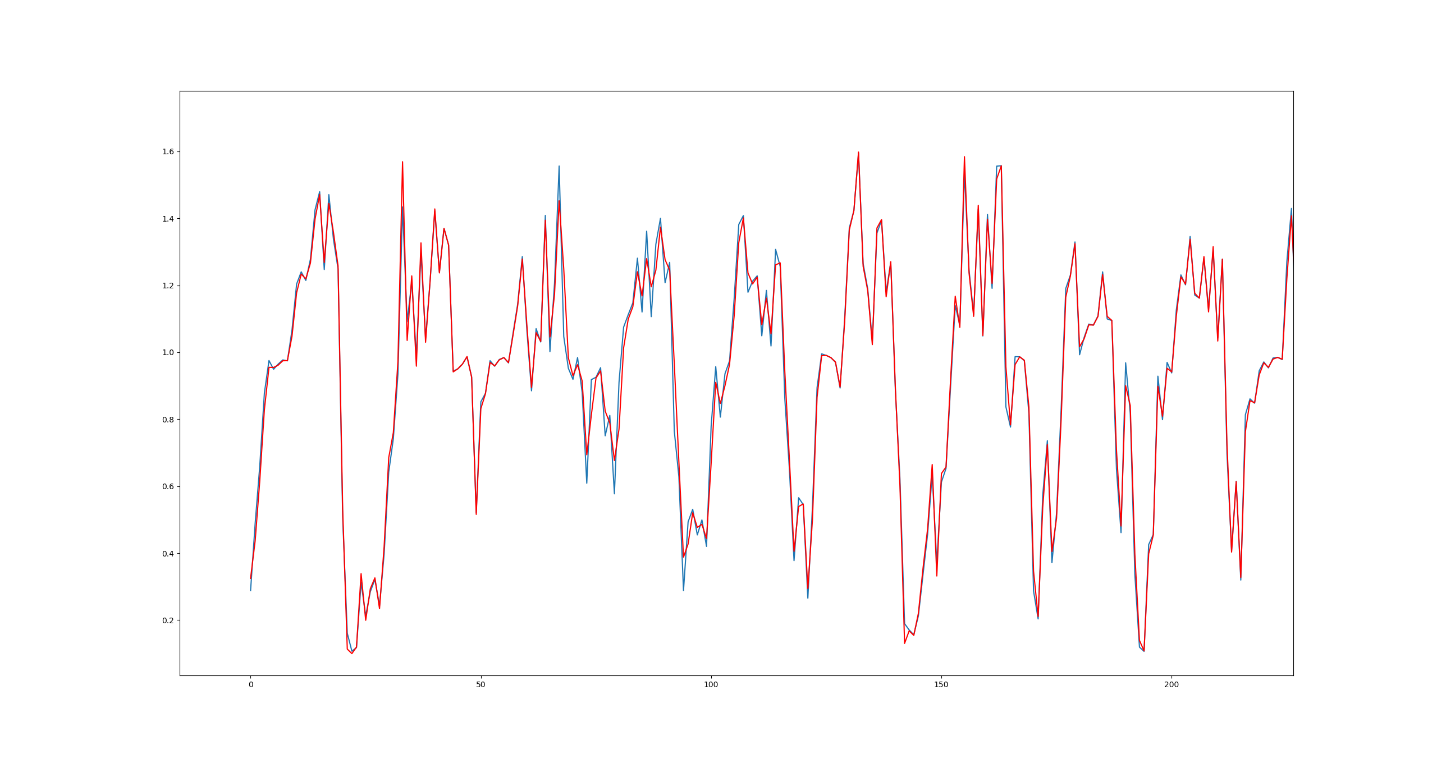
Une fois cela fait on prend un autre groupe de taille *n* et on évalue son résultat. En avançant notre sélection d’une valeur à la fois il est possible de faire de nombreuse évaluation. On évalue la RMSE (la moyenne des carres de l’erreur) pour comparer les différents modèles.

On cherche à prédire Kd. Nous avons les résultats suivants pour les ordres suivant :

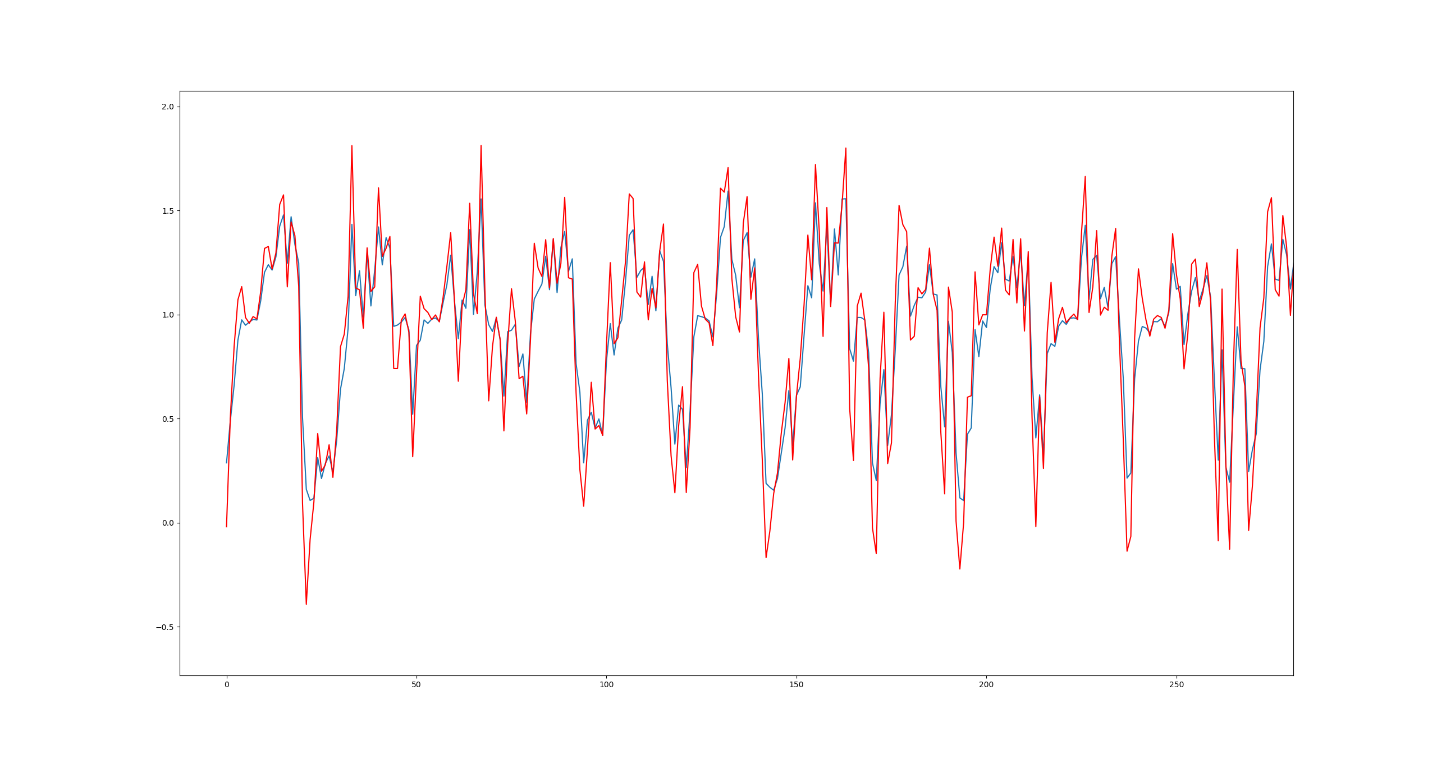
ARIMA (1, 0, 0) : RMSE: 0.0022 | std: 0.0004



ARIMA (1, 1, 0) : RMSE: 0.0013 | std: 0.0006



ARIMA (1, 2, 0) : RMSE: 0.0295 | std: 0.0040



Moufia : En bleu l’observation, en rouge la prédiction

Le fait d’avoir un ordre q de 1 créer un model moins biaise (RMSE plus faible) mais il est moins consistant (std plus grand). C’est deux valeur reste toutefois faible mais le model AR(1) est plus simple que ARIMA(1, 1, 0). Il semblerait que ce soit celui à privilégier.

## Améliorations possibles

La méthode ARIMA accepte des attributs en plus de la variable à prédire, il est possible qu’en fournissant plus de données on arrive à améliorer le résultat.

A l’heure actuelle les dernière 48h (moins les valeur nocturne) sont utilisée pour entrainer le model, en faisant varier cette valeur il est possible d’obtenir de meilleurs résultats (+) ou d’accélérer l’entrainement (-).

Il est possible d’utiliser des réseaux de neurones pour l’analyse de séries temporelle sous la forme de RNN et LSTM.

## Conclusion

Le but étant de faire de la prédiction de série temporelle j’étais parti sur la méthode ARIMA car il s’agit de celle le plus souvent utilise pour cette application. Mais après avoir cherche les paramètres optimaux il semblerai qu’il est possible d’utiliser la méthode AR(1) qui est beaucoup plus simple et rapide.