

Rapport projet Python



Kévin Lor
Quentin Garrido

E2 groupe 7
12/01/2018

I) Contexte

Le sujet général choisi est la production d'électricité au Royaume Uni. Nous nous sommes particulièrement intéressé à l'impact écologique de cette dernière. Nous avons donc choisi d'étudier deux données, le nombre de grammes de CO2 émis pour produire 1 kWh d'électricité et aussi les centrales électriques en Angleterre.

II) Les Histogrammes

1) Solution au problème

Afin d'obtenir les informations nécessaires pour les histogrammes nous avons eu recours à un API trouvé sur le site suivant: "<https://carbon-intensity.github.io/api-definitions/#carbon-intensity-api-v1-0-1>"

Ce dernier fournit des données en temps réel mais peut aussi donner des prédictions sur les émissions de CO2 jusqu'à 48h en avance. Les données présentes ne concerne que les émissions dues aux centrales électriques. Le site fournissait également des échantillons de codes selon le type de données recherchées (données d'une date précise ou sur une certaine période). Nous avons utilisé l'échantillon concernant le choix de données d'une période définie, ainsi nous représenterons des données prise toutes les 30 minutes entre le 12 septembre 2017 et le 19 décembre 2017.

Afin d'afficher les histogrammes, notre programme contenait 4 fonctions:

-def get_data(tab_urls):

Cette fonction prend en paramètre les urls des sites et permet de récupérer l'ensemble des données aux urls dans tab_urls. Ces dernières sont ensuite converties en json et nous récupérons les données présentent dans l'attribut data.

-def is_week_day(date):

Cette fonction prend en paramètre la date et la retourne si elle est un jour de la semaine (lundi à vendredi) ou non.

-def plot_scales(tab, fig, x_label, y_label, titre):

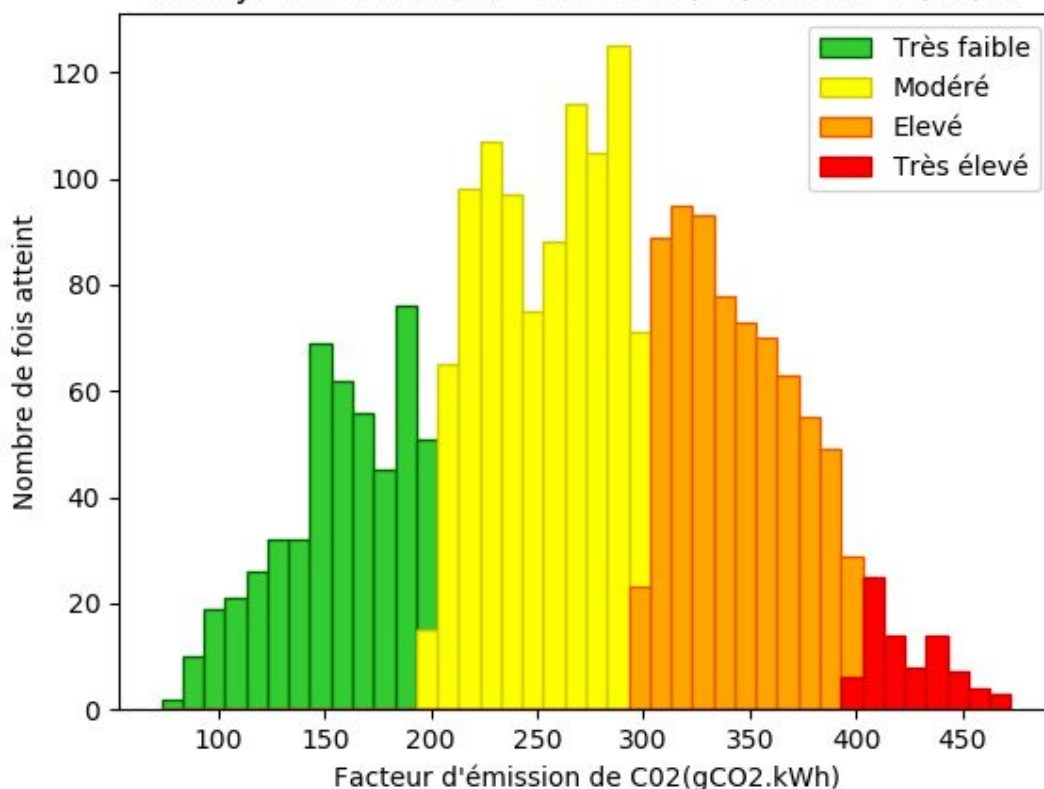
Cette fonction prend en paramètre le tableau des mesures, le type de figure du graphique, la description des axes x et y et le titre du graphique. C'est dans cette fonction que l'on va définir les différents intervalles du graphique. Les titres des axes et du graphique sont également définis dans cette fonction.

-def main():

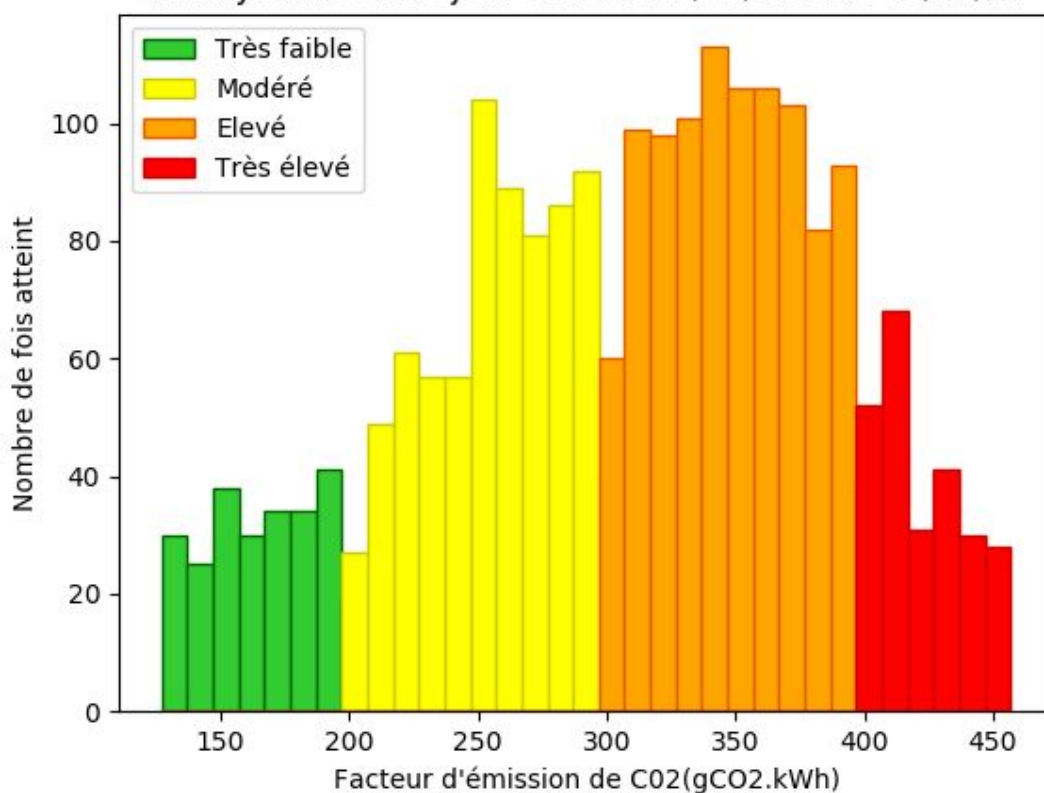
Cette fonction va récupérer les données, les traiter et les afficher à l'aide des différentes fonctions précédentes. Elle permet en outre de définir le jour (de 8h à 20h) et la nuit (de 20h à 8h), de différencier la semaine du week end et d'afficher l'ensemble des histogrammes avec leur axes respectifs.

Les histogrammes suivants représentent le nombre de fois qu'un facteur d'émission de CO₂ a été atteint durant une certaine période. En l'occurrence nous avons comparé 4 périodes, le jour, la nuit, la semaine, le week-end. Les différentes couleurs représentent les différents intervalles d'émission (vert pour moins de 200, jaune entre 200 et 300, orange entre 300 et 400 et rouge pour au dessus de 400).

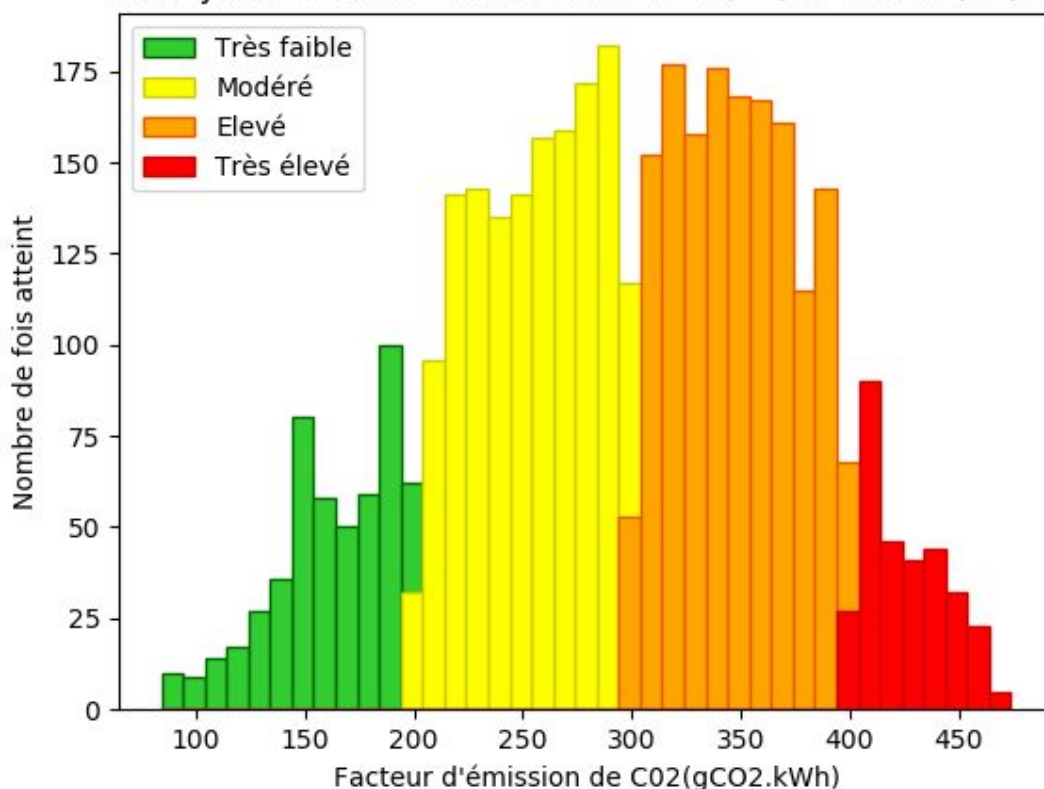
Impact environnemental de la production d'électricité
au Royaume Uni la nuit entre le 26/09/17 et le 19/12/17



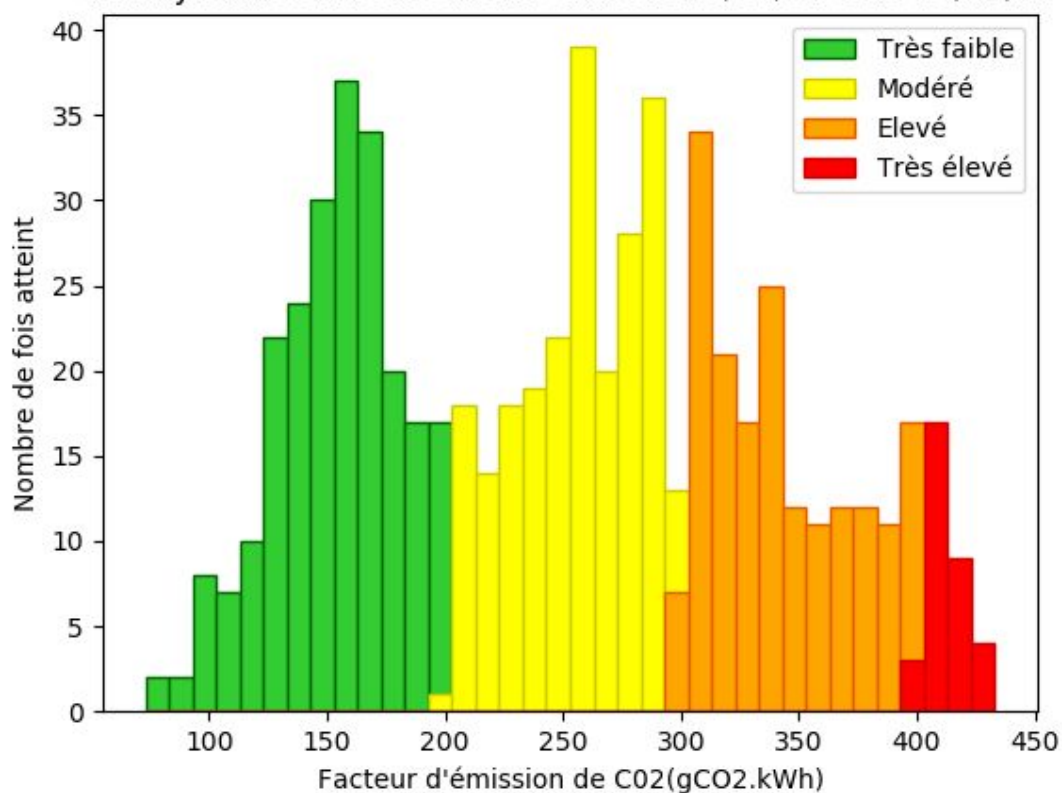
Impact environnemental de la production d'électricité
au Royaume Uni le jour entre le 26/09/17 et le 19/12/17



Impact environnemental de la production d'électricité
au Royaume Uni la semaine entre le 26/09/17 et le 19/12/17



Impact environnemental de la production d'électricité
au Royaume Uni le week-end entre le 26/09/17 et le 19/12/17



2)Interprétation des histogrammes

Nous avons décidé de tracer les valeurs prise en semaine, le week-end, le jour et la nuit afin des les comparer.

Sans surprise la production d'électricité est plus propre la nuit que le jour, la demande étant plus faible à cette période, les énergies non renouvelables sont moins sollicitées mais les énergies renouvelables fonctionnent comme en temps normal car nous ne pouvons pas contrôler les éléments.

Nous pouvons constater la même chose entre la semaine et le week-end pour des raisons similaires. Le week-end la majeure partie de la population est chez elle et donc utilise plus d'électricité, donc les centrales électriques sont plus sollicitées d'où la différence de propreté entre les deux.

III) Notre carte

1)Solution au problème

Pour la représentation géolocalisée nous avons parsé la page wikipédia suivante : ["https://en.wikipedia.org/wiki/List_of_power_stations_in_England"](https://en.wikipedia.org/wiki/List_of_power_stations_in_England).

Nous n'avons gardé que les stations encore ouvertes et les 20 plus grandes de celles-ci, pour avoir une carte plus clair et moins de disparité d'importance entre les stations.

Pour la représenter nous avons utilisé les 8 fonctions suivantes:

- def parse_html(html_data):

Cette fonction permet de récupérer l'ensemble des tables contenant les données utiles de la page web. Pour cela nous avons fait appel à la bibliothèque BeautifulSoup qui permet de lire chaque ligne et colonne de manière plus simple qu'en utilisant HTMLParser, le fonctionnement sous jacent est le même mais BeautifulSoup nous a permis de réaliser la tâche plus facilement. L'ensemble des données sera stockés dans une liste qui correspondra à une station. puis toutes les listes vont ensuite être stockées dans une liste différente.

- def filter_coordinates(listes_stations):

Cette fonction prend en paramètre une liste de stations et permet de structurer les coordonnées et de les convertir au format (°N,°E) pour pouvoir les utiliser.

- def filter_open(listes_stations):

Cette fonction prend en paramètre une liste de stations et retourne celles qui sont encore en activité.

- def get_useful_data(liste):

Cette fonction prend en paramètre une liste dans laquelle sont présentes toutes les données et en retourne une où seuls sont présents les éléments utiles à notre projet (Coordonnées, type, capacité de production).

- def filter_out_small_plants(listes, cap):

Cette fonction prend en paramètre la liste des stations et les “cap” plus grandes stations (en fonction de leur capacité), par exemple si cap=20 on aura les 20 plus grandes stations.

- def get_data_plants():

Il s’agit de la fonction principale qui va récupérer l’ensemble des données et les trier/ordonner grâce aux fonctions précédentes.

Nous avons ensuite créé un autre fichier map_uk.py qui va se charger d’afficher la carte.

Ce fichier contient l’ensemble des “MARKERS” qui vont servir de symboles qui vont être affichés sur la carte. De plus le fichier contenait 4 fonctions:

-def create_legend(marker):

Cette fonction prend en paramètre les marqueurs et va créer la légende à partir de ces derniers.

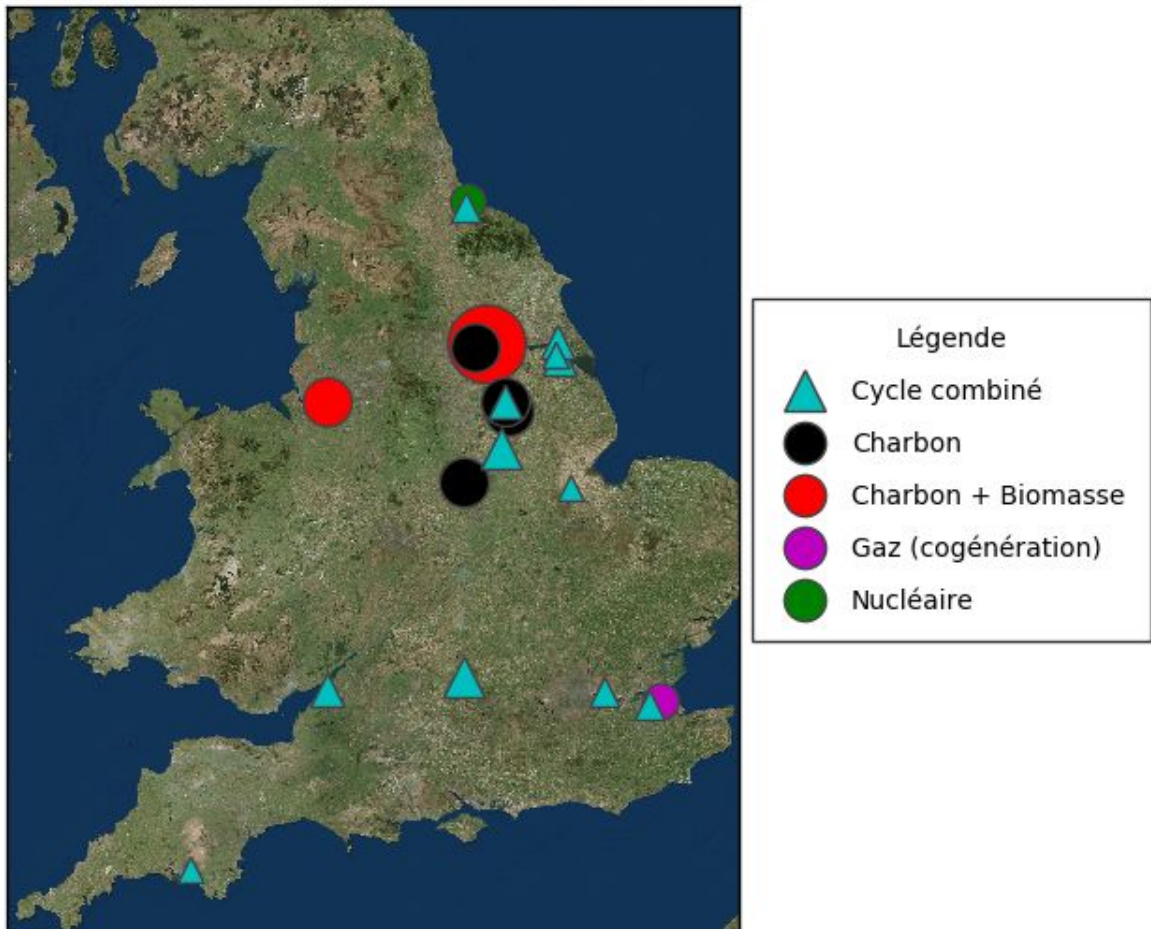
-def scale_size(data):

Cette fonction prend en paramètre les données des stations et les met à une bonne échelle afin d’obtenir un bon affichage.

-def main():

Il s'agit de la fonction principale qui va récupérer puis trier les données pour les afficher. Pour cela nous allons utiliser l'extension Basemap qui permet de représenter des données sur une carte. Ces données sont triées du plus grand au plus petit afin d'afficher les plus grands symboles en premiers afin de ne pas masquer les plus petits. La fonction va ensuite ajouter un fond ainsi qu'un titre à la carte.

20 plus grandes centrales électrique d'Angleterre



2)Interprétation de la carte

Nous pouvons voir que les plus grosses centrales électriques sont des centrales à cycle combiné (gaz) et charbon/charbon + biomasse, le nucléaire étant bien moins représenté. Cela montre bien que la production d'électricité au Royaume-Uni est en retard sur d'autres pays comme la France où le nucléaire (qui pollue globalement moins que le charbon) est plus représenté.

En corrélant la carte avec les histogrammes vus précédemment on comprend encore mieux les disparités dans la production, car quand l'électricité est produite par des énergies fossiles elle l'est par celles n'étant pas forcément le plus propre, malgré cela les centrales au gaz polluent bien moins que celles au charbon.