VISUALISATION DE DONNÉES
- L3 MATH, ISD, TP 1 -

Ce TP vise à initier :
- Aux packages matplotlib et seaborn de python pour la construction de graphiques figés.
- Au package plotly de python pour les graphiques dynamiques, au format web.

Nous verrons par la suite la manière de construire des cartes facilement avec des formats équivalents.

Chercher à produire une visualisation parfaite du premier coup est illusoire. Il est beaucoup plus réaliste d'améliorer graduellement une représentation graphique afin, petit à petit, de mettre en avant les effets de structure dans un jeu de données.
Nous allons donc commencer par nous représenter la distribution des production d'électricité dans les différentes communes de la région rénaise. Pour cela nous allons produire rapidement un barplot puis l'améliorer graduellement.

Environnement de travail

Nous allons travailler avec le langage python sous google colaboratory qui permet d'utiliser des notebooks. Pour ouvrir un nouveau notebook, cliquer sur le lien google colaboratory et choisir d'ouvrir un nouveau notebook (bouton bleu en bas à gauche). Le notebook sera automatiquement sauvegarder dans votre drive google [1].

Données

Un sous-ensemble des données de Rennes Métropole En Accès Libre a été mis à disposition pour faciliter l'import. Seules les colonnes qui servent à cet exercice ont été conservées.
Les productions d'énergie photovaltaïque et éoliennes sont produites à une échelle relativement fine : par commune, année et domaine de tension.

Premières productions graphiques avec l'API Matplotlib de Pandas

Les données comportent plusieurs dimensions pouvant faire l'objet d'une analyse statistique. Il est donc nécessaire dans un premier temps de synthétiser celles-ci par des agrégations afin d'avoir un graphique lisible. Nous proposons d'agréger les données de façon à obtenir une production moyenne par commune.

1. Importer les données de production d'énergie. Vous pouvez utiliser l'url https://perso.univ-rennes1.fr/valerie.monbet/ISD/production-photovoltaique-tp1.csv. Le dataframe obtenu sera nommé production.
   Combien de lignes et de colonnes comportent le dataframe production ?
   Quelles sont les noms de colonnes ?

1. si vous n'avez pas de compte google, vous pouvez travailler avec kaggle : voir la page moodle du cours

A quoi correspond une ligne ?

2. En utilisant les fonctions groupby et sum de pandas, agréger les données par commune et année. Le dataframe obtenu sera nommé productioncumulee.
3. En utilisant les fonctions groupby et mean de pandas, obtenir une table des productions annuelles maoyennes par commune. Le dataframe obtenu sera nommé productionmoyenne.
4. En premier lieu, sans se préoccuper des éléments de style ni de la beauté du graphique, créer la structure du barplot (diagramme en batons) de la figure 1 (à gauche). Notez qu'on travaille avec la production photovoltaique moyenne annuelle (ie le dataframe productionmoyenne).

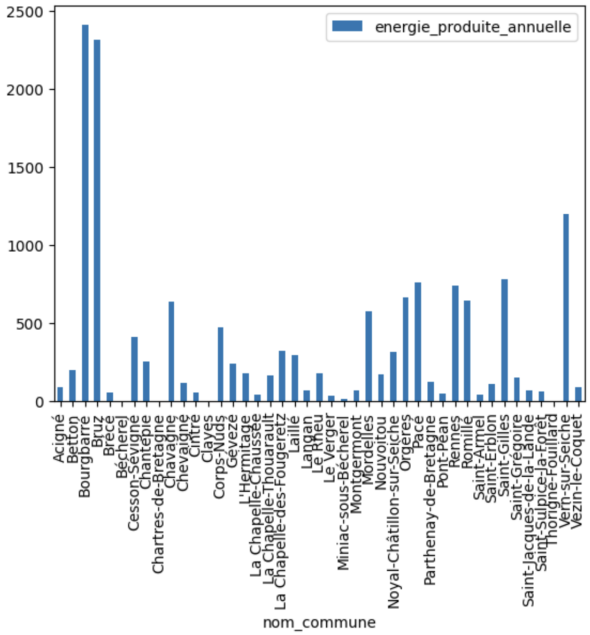

FIGURE 1 – - Production Photovoltaique moyenne (MWh) du bassin rennais, sans travail sur le style.

5. Pour préparer le travail de la seconde figure (Fig. 2), on ne conserve que les 10 communes ayant la production photovaltaique annuelle la plus importante. On pourra utiliser la fonction sort du module pandas Sans se préoccuper des éléments de style ni de la beauté du graphique, créer la structure du barplot pour les 10 communes sélectionnées.
6. Comme pour la question précédente, créer un barplot pour reproduire la figure 3.

Utiliser directement Seaborn

L'objectif est d'obtenir la figure 4.

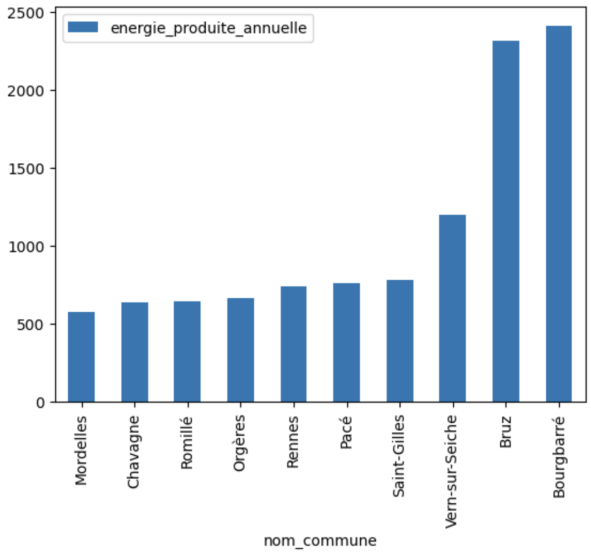

FIGURE 2 – Production Photovoltaique moyenne (MWh) du bassin rennais, sans travail sur le style.

1. Réinitialiser l'index du dataframe best10 pour avoir une colonne "Nom de commune". Réordonner les données de manière décroissante pour obtenir un graphique ordonné dans le bon sens avec seaborn.
2. Refaire le graphique précédent avec la fonction barplot de seaborn. Pour contrôler la taille du graphique vous pouvez utiliser les arguments `height` et `aspect`. On obtient une figure équivalente à la figure 3.
3. Ajouter les titres des axes et le titre du graphique.
4. Essayez de colorer en rouge l'axe des x. Vous pouvez pré-définir un style avec `sns.set_style("ticks", {"xtick.color": "red"})` [2]

Lillipop chart avec matplotlib

En communication, il est important d'utiliser les graphismes à la mode. En suivant l'approche pas à pas des questions précédentes, reproduire le lillipop chart de la figure 6.

Séries temporelles

On va maintenant se concentrer sur l'évolution temporelle de la production d'énergie photovoltaique entre 2011 et 2022.

1. Dans un premier temps, on s'intéresse à la production globale à l'échelle du bassin rennais. En utilisant de nouveau les commandes groupby et sum, agréger la production d'énergie photovaltaique pour obtenir une table comme ci-dessous (ici on n'affiche que les premières lignes)

2. attention : les options de style restent actives sous python ; pour les désactiver il faut penser à les réinitialiser au valeurs par défaut.

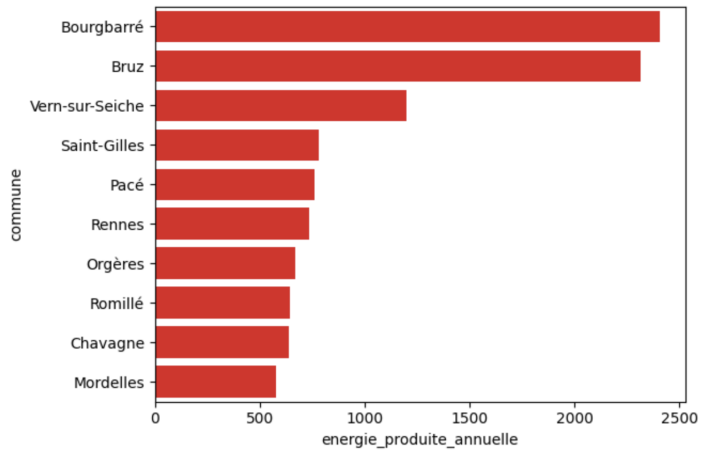

FIGURE 3 – Production Photovoltaique moyenne (MWh) du bassin rennais.

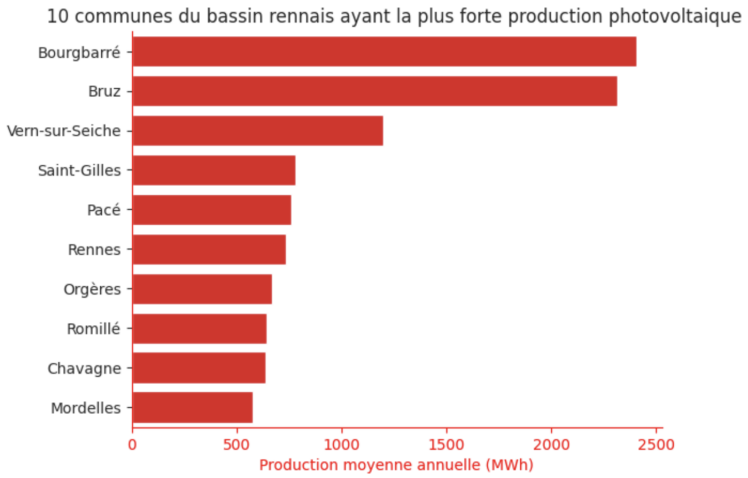

FIGURE 4 – Production Photovoltaique moyenne (MWh) du bassin rennais, figure tracée avec seaborn avec options de style.

	nb_sites	energie_produite	annee
annee			
2011	679.0	5449.530934	2011
2012	716.0	8715.920945	2012
2013	896.0	10180.062945	2013
2014	958.0	11851.479904	2014
2015	1010.0	13465.478971	2015

2. Tracer l'évolution temporelle de la production d'énergie photovaltaique avec les fonctions lineplot de seaborn et fill between de matplotlib , voir Fig. ??.

Des graphiques dynamiques avec Plotly

L'inconvénient des figures avec seaborn est que celles-ci ne permettent pas d'interaction avec le lecteur. Toute l'information doit donc être contenue dans la figure ce qui peut la rendre

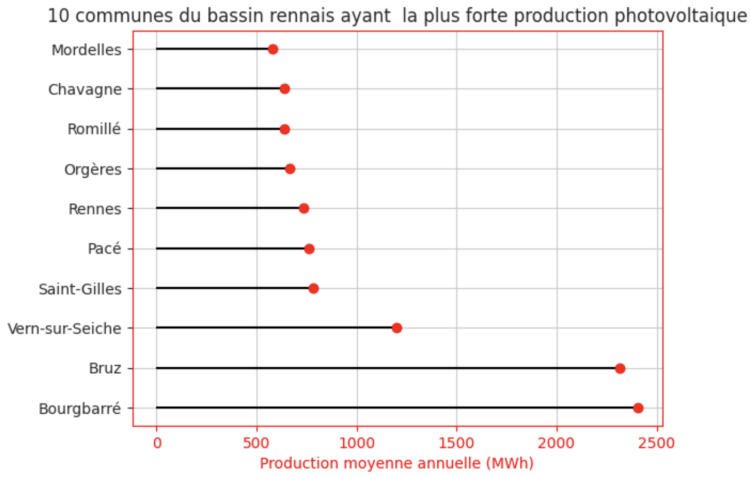

FIGURE 5 – Production Photovoltaique moyenne (MWh) du bassin rennais, lillipop chart.

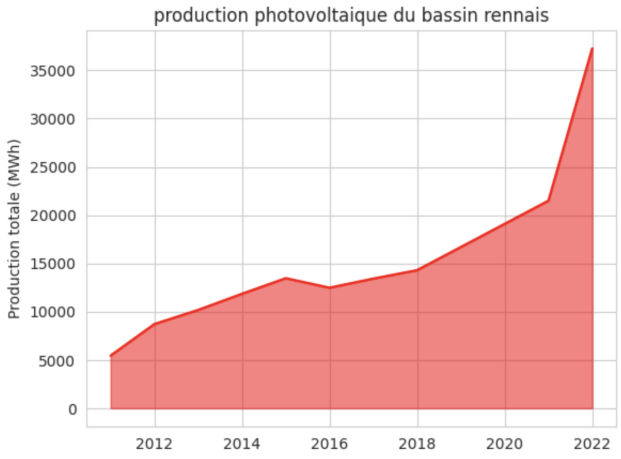

FIGURE 6 – Production Photovoltaique totale (MWh) du bassin rennais.

difficile à lire. Si la figure est bien faite, avec différents niveaux d'information, cela peut bien fonctionner.

Il est néanmoins plus simple, grâce aux technologies web, de proposer des visualisations à plusieurs niveaux. Un premier niveau d'information, celui du coup d'oeil, peut suffire à assimiler les principaux messages de la visualisation. Ensuite, un comportement plus volontaire de recherche d'information secondaire peut permettre d'en savoir plus. Les visualisations réactives, qui sont maintenant la norme dans le monde de la dataviz, permettent ce type d'approche : le lecteur d'une visualisation peut passer sa souris à la recherche d'information complémentaire (par exemple les valeurs exactes) ou cliquer pour faire apparaître des informations complémentaires sur la visualisation ou autour.

L'objectif est de reconstuire le premier diagramme en barre rouge (fig. 4) avec Plotly.

1. Réalisez le graphique en utilisant la fonction adéquate avec plotly.express

Pour la couleur rouge, vous pouvez utiliser l'argument color discrete sequence.
Ne pas oublier de nommer les axes
Pensez à la couleur du texte de l'axe inférieur

2. Tester un autre thème, à fond blanc par exemple. Pour les couleurs, faire un groupe stockant les trois plus fortes valeurs puis les autres.