

Projet Big Data



# Introduction

Dans le cadre de notre projet, notre commanditaire, ici Alex Lima nous a demandé de réaliser différentes analyses provenant au minimum de 2 sources de données différentes. Nous avons donc choisi :

* une source de données provenant de PIB par habitant d’un pays qui est estimé en milliers d’euros
* une autre source de données provenant des émissions de CO2 émises en tonne/habitant.

Nous allons voir dans ce document, comment nous nous sommes répartis les tâches, nos différentes problématiques, nos solutions, nos tests et nos scripts…



Table des matières

[Introduction 2](#_Toc119576275)

[I- Définition du besoin/architecture 4](#_Toc119576276)

[II- Répartitions des tâches 6](#_Toc119576277)

[III- Problématiques 6](#_Toc119576278)

[IV- Solutions 6](#_Toc119576279)

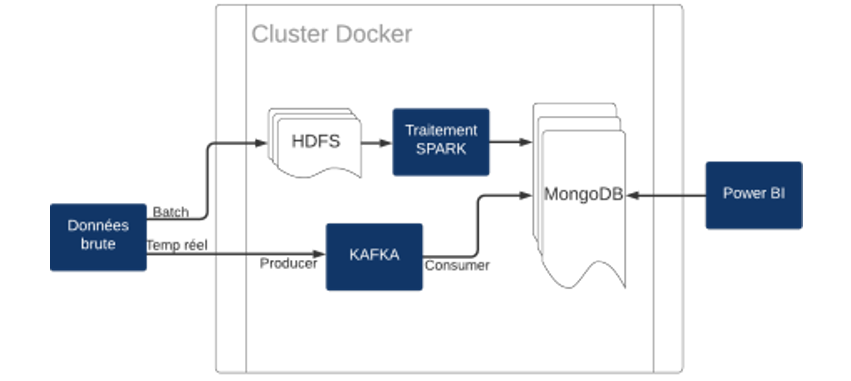
[V- Les différents scripts 8](#_Toc119576280)

[VI- Les résultats 11](#_Toc119576281)

[Conclusion : 13](#_Toc119576282)

# Définition du besoin/architecture

Le besoin consiste à traiter des données provenant de différentes sources, 2 minimum. Pour réaliser cela, nous nous sommes appuyés sur l’architecture suivante :



**Comme première source de données**, nous avons choisi le PIB par habitant d’un pays selon une année, qui sera estimé en milliers d’euros.

Une image contenant table

Description générée automatiquement

**Comme deuxième source de données**, nous avons choisis les émissions de CO2 par habitant en fonction du pays selon une année, qui seront estimées en tonne/habitant.

Une image contenant table

Description générée automatiquement

Nous avons choisi d’utiliser des données « Batch » donc en passant par « l’HDFS », un « traitement Spark », puis un outil de visualisation qui est « Power Bi Desktop ».

Tous nos scripts/tests ont été réalisés sur le notebook de Jupyterlab.

Les différentes étapes sont :

* Récupérer les 2 datasets en format csv
* corriger les csv
* stocker les 2 datasets dans l’HDFS
* effectuer les tests sur le notebook de jupyterlab
* effectuer le scripte python en ouvrant une nouvelle page sur jupyterlab
* ouvrir Power BI Desktop et récupérer le csv (part-…)
* générer grâce au scripte python et analyser grâce au graphique souhaité les données.

# Répartitions des tâches

|  |  |  |
| --- | --- | --- |
| Wassil | Semy | Yoni |
| Initialisation/traitement avec HDFS | Scrapping des données | Traitement spark |
| Traitement spark | Réception des datasets d’origine sous format csv | Analyse sur jupyterlab/ Initialisation/traitement avec HDFS |
| Assemblages des VM sur le même container/Analyse sur jupyterlab | Modification csv de base pour bien adapter à notre besoin | Modification csv de base pour bien adapter à notre besoin |
| Modification csv de base pour bien adapter à notre besoin | Initialisation/traitement avec HDFS | Assemblages des VM sur le même container |
| Initialisation outil de visualisation (Power Bi Desktop) | Initialisation outil de visualisation (Power Bi Desktop) | Initialisation outil de visualisation (Power Bi Desktop) |

# Problématiques

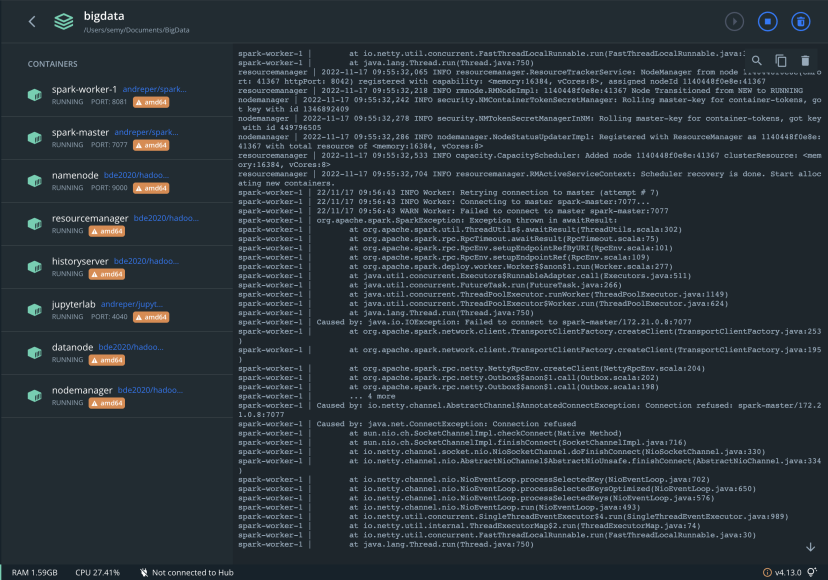
Nous avons eu différentes problématiques :

* au niveau de la mise en place des VM dans le même container
* de l’assemblement des 2 csv datasets d’origine en 1 seul et même csv (Script python)
* du choix de l’outil de visualisation
* de la modification sur Excel directement des 2 datasets donc de la réorganisation des fichiers
* des problèmes d’importation de bibliothèques (pour le script python)

# Solutions

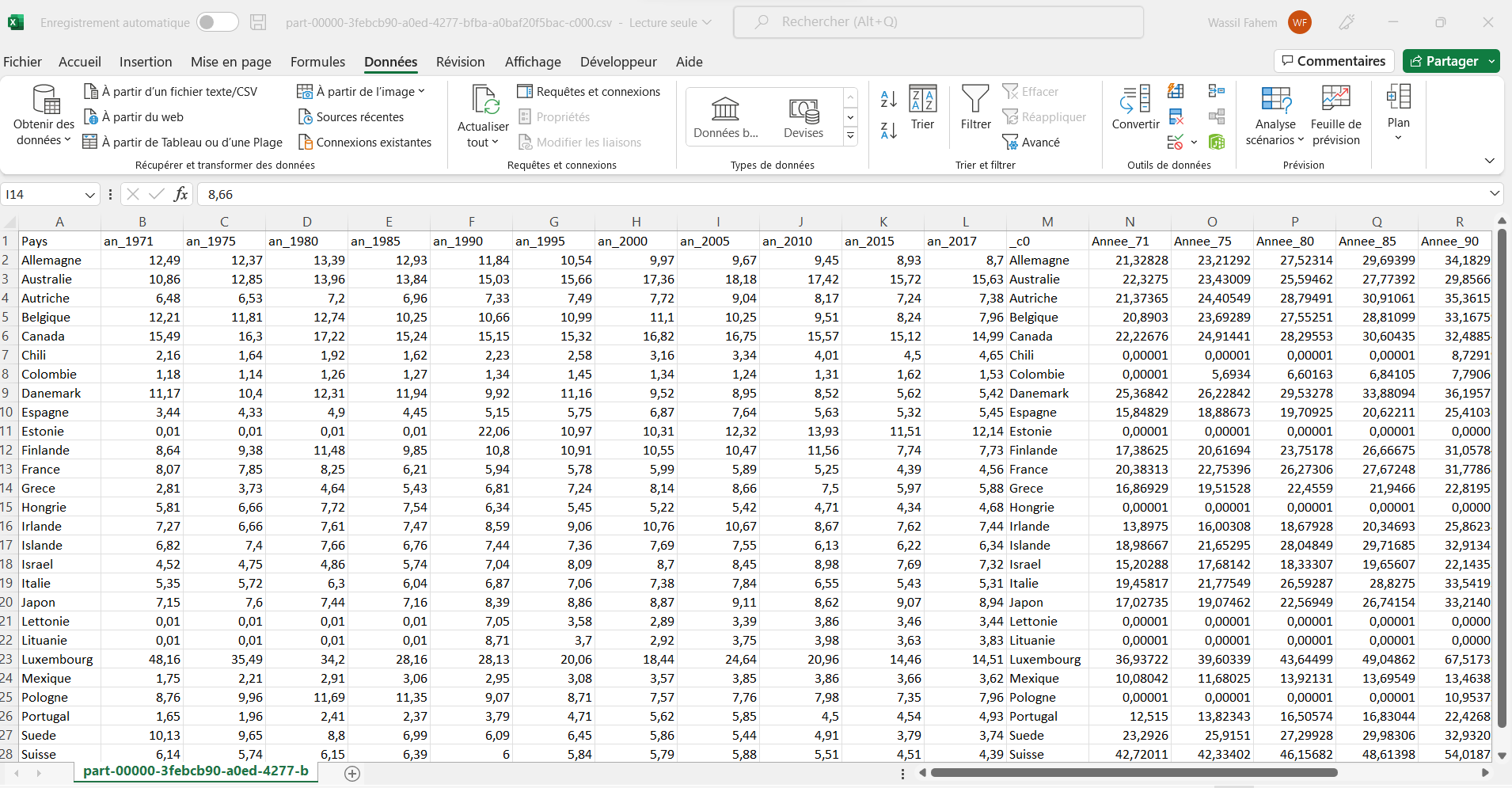
Nos solutions ont été de bien réfléchir et comprendre ce que l’on fait et ce que l’on souhaite faire. Nous avons regardé sur des forums en ligne, des vidéos sur YouTube … Dans le but de régler nos différents problèmes.

Pour ce qui est de l’assemblage des VM dans un container :





Voici le part-00000, générer grâce au script avec les 2 datasets de bases, stocké en 1 csv qui est ci-dessous.



# Scripts

Commande pour copier les 2 datasets du local vers la VM principale de Hadoop ; le namenode.

Une image contenant texte

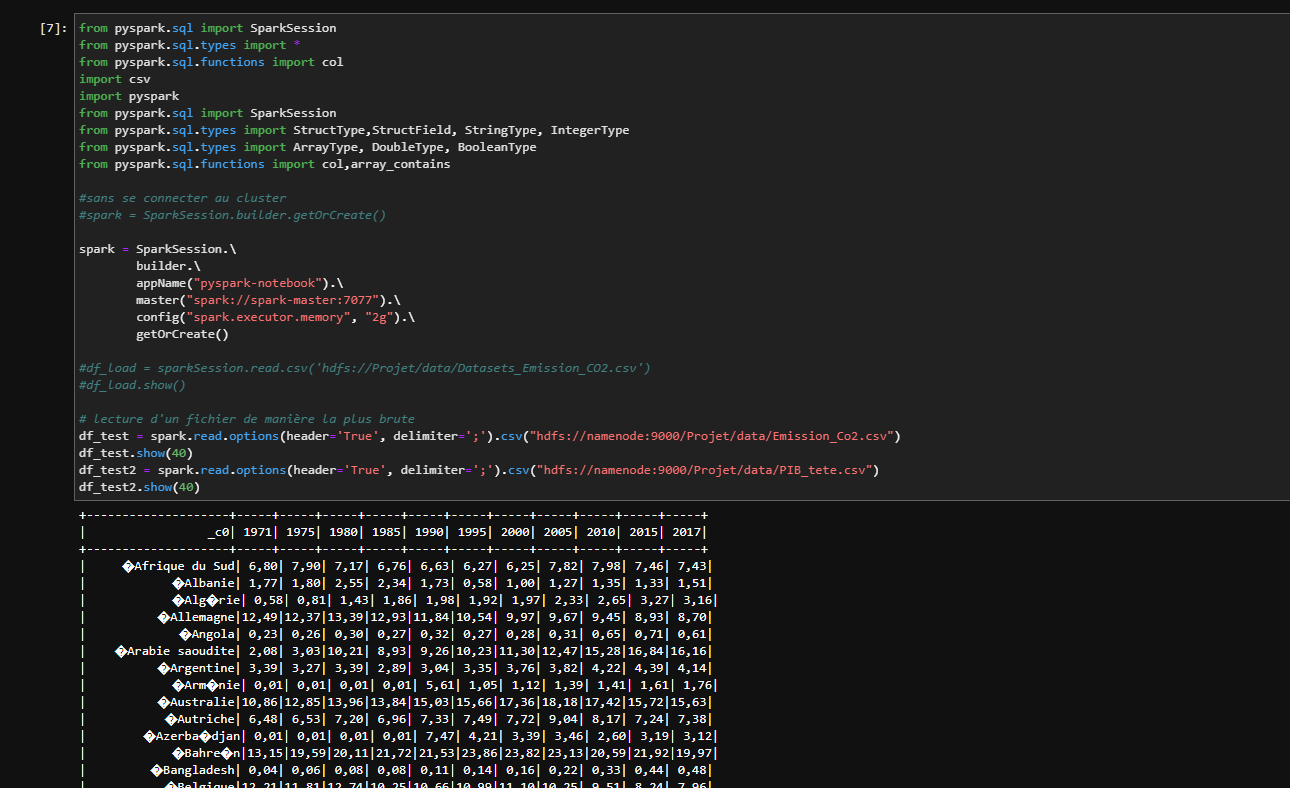
Description générée automatiquement

Ici, nous pouvons voir que nous avons stocké les 2 datasets, dans l’HDFS, dans le dossier Projet puis le sous dossier data.

Une image contenant texte

Description générée automatiquement

Voici le script sur jupyterlab qui permet d’importer les différentes librairies python externe qu’il a fallu installer dans un premier temps, ensuite un chargement d’une session spark, puis la lecture des 2 datasets qui sont stockés dans l’HDFS.



Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

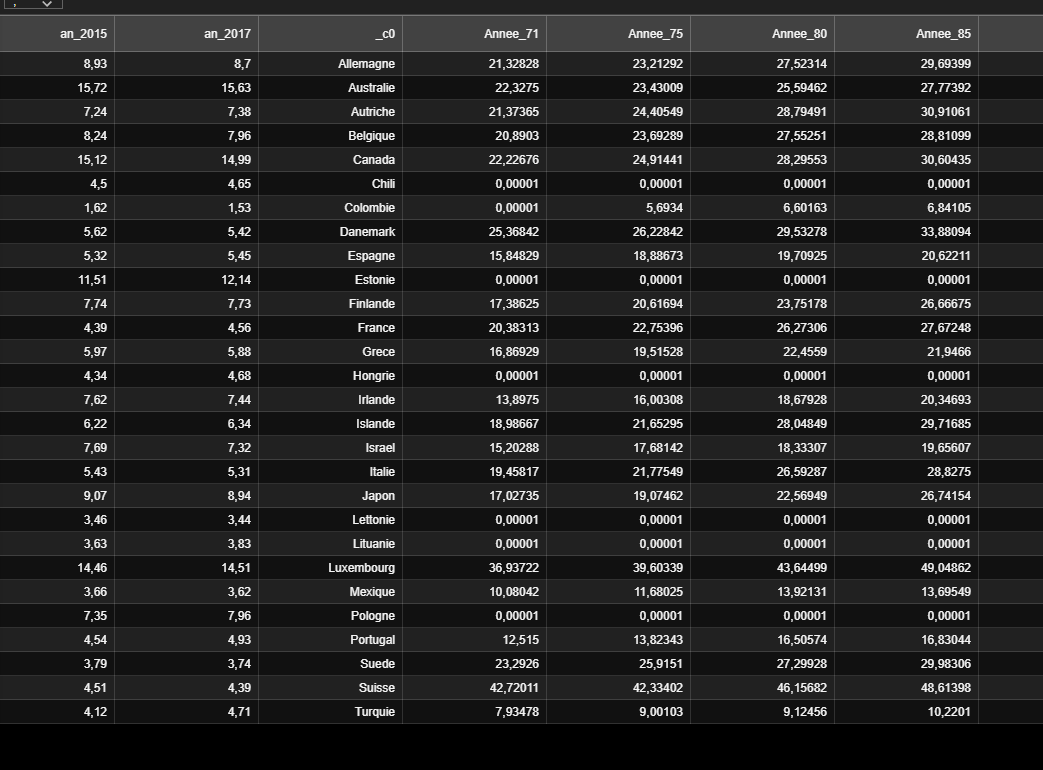
En rajoutant les 2 dernières lignes, nous pouvons joindre nos 2 datasets de base au sein d’un même csv, qui sera stocké dans « result\_test ».

Une image contenant texte

Description générée automatiquement

Une image contenant texte, capture d’écran, ordinateur

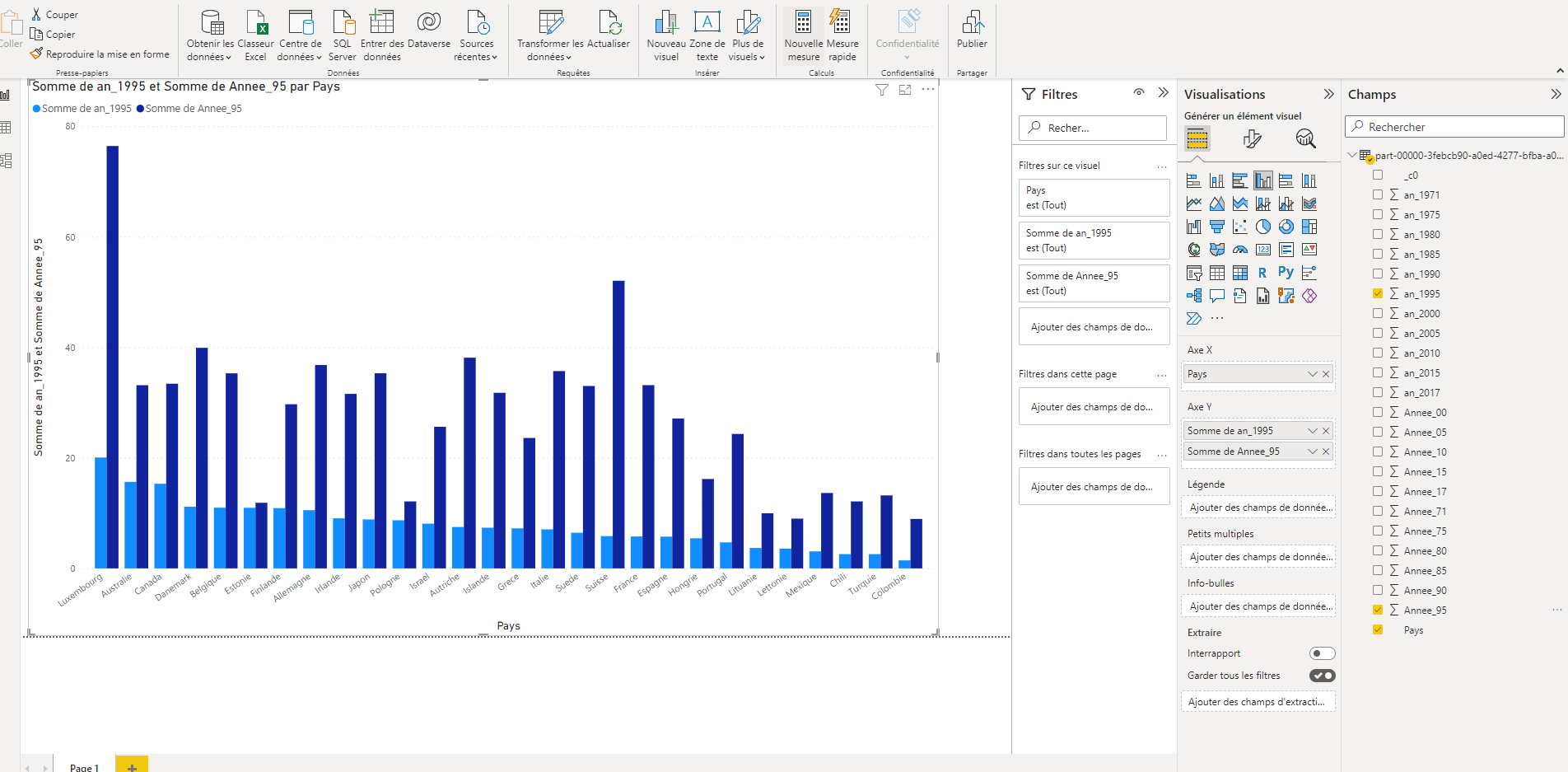
Description générée automatiquement



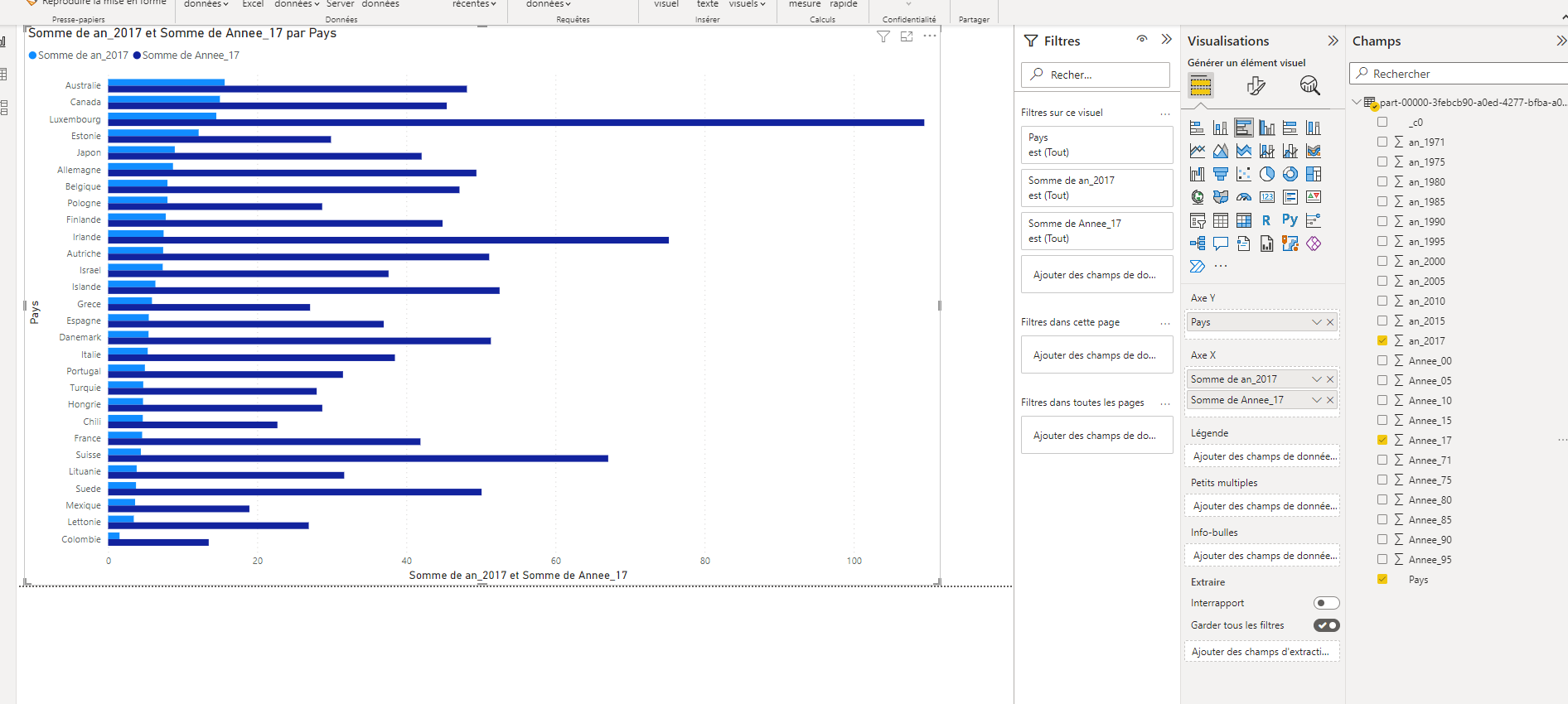
Nous pouvons voir ci-dessus que le « part », quand on l’ouvre il y a bien nos 2 datasets en 1.

# Les résultats

Voyons maintenant ce qui se passe quand on insère notre csv (part) dans PowerBi Desktop. (Année…= PIB) (an…= émissions Co2).

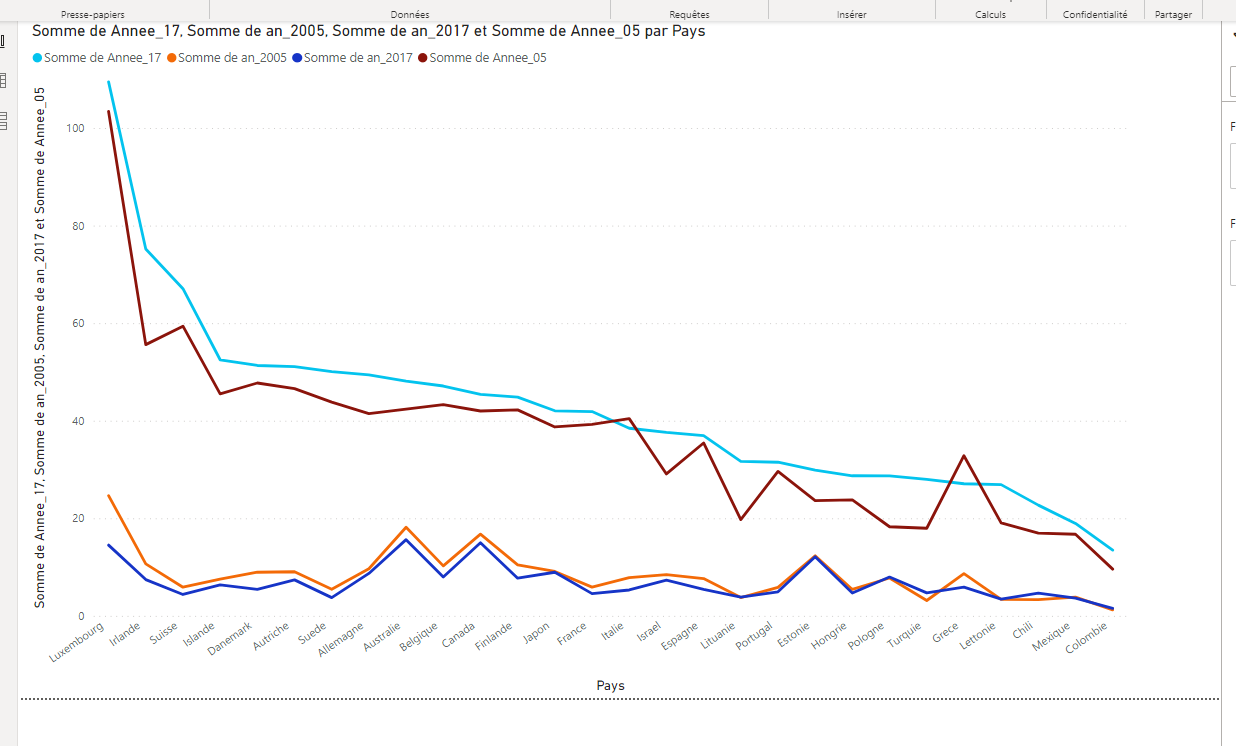


Nous avons donc ci-dessus, avec le part-00000 que l’on peut constater dans la partie « Champs » tout à droite. Nous avons un graphique en barre avec la somme de l’année 1995 par pays pour le dataset de base PIB qui est de couleur bleu foncé sur le graphique et de couleur bleu clair c’est le dataset émissions Co2.



Pour ce graphique ci-dessus, nous avons choisis d’analyser l’année 2017. La légende pour la couleur est la même que la précédente.

Pour le graphique ci-dessous, nous avons décider de voir la différence entre l’année 2005 et 2017. Nous pouvons voir qu’il y a une légère baisse pour l’année 2017 par rapport à 2005 pour certains pays concernant les émissions et qu’il y a une hausse pour presque tous les pays pour le PIB en 2017 par rapport à 2005.



# Conclusion :

Par conséquent, ce projet a été très enrichissant pour nous, nous avons pu découvrir différentes notions tels que le traitement Spark, s’avoir se diriger dans l’invite de commandes sur les différentes VM, sur le filesystem également, approfondissement de python, association avec du SQL, initialisation d’un outil de visualisation ; ici power bi Desktop.

L’axe d’amélioration que nous pourrons avoir serait d’augmenter le nombre de sources de données par exemple pour traiter davantage des données et en sortir des résultats encore plus intéressants.



