

Rapport projet Visualisation de Donnée

Mathiam Faye - Mbaye Thilor Sene

Introduction

Ce rapport est rédigé dans le cadre du projet final du cours de Visualisation de De Données en Master 1 Sciences de Données et Applications.

Dans ce projet nous avons utilisé des données sur les prix des céréales au Sénégal, notamment leurs évolutions au cours du temps de 2000 à 2020. Les données sont issues de la plateforme <https://data.humdata.org/> plus connu sous le nom HDX (Humanitarian Data EXchange). C'est un open source de partage de données humanitaire créé en 2014 son but de faciliter l'accès et l'exploitation de plus de 18 000 datasets sur 253 pays à travers le monde.

Les données recueillis apres traitements sous panda (Pandas) sont représentées et visualisées dans le site <https://thilorsene.github.io/finalvisualisation/> développé à cette occasion.

Le processus ainsi que les technologies et langage utilisés vous seront présenté sur ce rapport.

Conception

1. Choix du sujet :

La consigne étant de choisir des données concernant les pays de la CEDEAO (Communauté Economique Des Etats Afrique de l'Ouest), nous avons choisi d'orienter notre travail sur le Sénégal qui est pays de cette communauté et de trouver des données sur les céréales qui sont essentiels à notre survie car constituant la nourriture de base dans l'ensemble des pays de la CEDEAO. Sur ces céréales on a collecté des sur la progression du prix des 4 plus consommés au Sénégal : le riz, le maïs, le sorgho et le mil.

2. Traitement des Données :

Obtenu sous le format CSV (Comma Separated Values) depuis le site de HDX certaines colonnes du dataset sont incomplètes, d'autres ne nous intéresse pas dans ce projet.

Nous avons utilisé Pandas une bibliothèque python pour la DataScience qui offre une large panoplie de fonctions simples à utiliser.

Nous avons changé le format des dates et ne conserver que l'année afin de pouvoir y appliquer une fonction d'agrégation.

Projet Visualisation - Jupyter Notebook - Mozilla Firefox

127.0.0.1:5500/carte.html x Desktop/M1/51/ x Projet Visualisation - Jupyter Notebook - Mozilla Firefox x Traitement Données CSV x

localhost:8888/notebooks/Desktop/M1/51/Projet Visualisation.ipynb

jupyter Projet Visualisation Last Checkpoint: 04/14/2020 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [1]: import pandas as pd
In [2]: prix = pd.read_csv('food_prices.csv')
        prix2 = pd.read_csv('food.csv')
In [3]: prix
```

Out[3]:

	#date	#item+name	#item+unit	#item+type	#value	#currency	#country+name	#adm1+name	#adm1+code	#name+market	NaN	#item+code	NaN	NaN
0														
1	2007-01-15	Maize (local) - Retail	KG	cereals and tubers	160.0	XOF	Senegal	Kedougou	1374	Kedougou	419.0	56	15.0	5.1
2	2007-03-15	Maize (local) - Retail	KG	cereals and tubers	100.0	XOF	Senegal	Kedougou	1374	Kedougou	419.0	56	15.0	5.1
3	2007-04-15	Maize (local) - Retail	KG	cereals and tubers	200.0	XOF	Senegal	Kedougou	1374	Kedougou	419.0	56	15.0	5.1
4	2007-05-15	Maize (local) - Retail	KG	cereals and tubers	200.0	XOF	Senegal	Kedougou	1374	Kedougou	419.0	56	15.0	5.1
...
2013	2013-01-15	Maize (local) - Retail	KG	cereals and tubers	160.0	XOF	Senegal	Kedougou	1374	Kedougou	419.0	56	15.0	5.1

Projet Visualisation - Jupyter Notebook - Mozilla Firefox

127.0.0.1:5500/carte.html x Desktop/M1/51/ x Projet Visualisation - Jupyter Notebook - Mozilla Firefox x Traitement Données CSV x

localhost:8888/notebooks/Desktop/M1/51/Projet Visualisation.ipynb

jupyter Projet Visualisation Last Checkpoint: 04/14/2020 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [38]: tst=prix2.date
         tst=list(tst)
         type(tst)
         tst[0][0:4]
Out[38]: '2007'
```

In [39]: for i in range(len(tst)):
 tst[i]=int(tst[i][0:4])

In []:

In [40]: prix2.date=tst

In [56]: prix2.date

Out[56]:

0	2007
1	2007
2	2007
3	2007
4	2007
...	...
29911	2012
29912	2012
29913	2012
29914	2012
29915	2012

Name: date Length: 29916 dtype: int64

Après le changement de format la fonction d'agrégation regroupe les enregistrements par céréales et année pour calculer le prix moyen calculer la moyenne des prix par année afin d'obtenir une visualisation plus optimale.

```

In [58]: a = prix2.loc[:,['date','cnnname','price','adnname']]

In [73]: a=a.groupby(['date','adnname','cnnname']).mean()

In [74]: type(a)
Out[74]: pandas.core.frame.DataFrame

In [78]: a
Out[78]:

```

	date	adnname	cnnname	price	
2000		Dakar	Maize (imported) - Retail	153.137500	
			Millet - Retail	127.711800	
			Rice (imported) - Retail	225.309025	
			Sorghum - Retail	126.503475	
2001		Dakar	Maize (imported) - Retail	164.703480	
...	
2020		Thies	Sorghum - Retail	253.333340	
			Ziguinchor	Maize (local) - Retail	266.666667
			Millet - Retail	275.000000	
			Rice (imported) - Retail	325.000000	

Et pour finir la création des fichiers CSV contenant les prix chaque céréale.

```

In [9]: b=prix.loc['Maize (imported) - Retail']

In [11]: b.to_csv('Mais.csv')

In [12]: b=prix.loc['Maize (local) - Retail']

In [13]: b.to_csv('Maislocal.csv')

In [14]: b=prix.loc['Millet - Retail']

In [15]: b.to_csv('Millet.csv')

In [17]: b=prix.loc['Rice (imported) - Retail']

In [18]: b.to_csv('Riz.csv')

In [19]: b=prix.loc['Rice (local) - Retail']
b.to_csv('Rizlocal.csv')

In [20]: b=prix.loc['Sorghum - Retail']
b.to_csv('Sorgho.csv')

In [ ]:

```

3. Création du fichier GeoJson pour la carte :

Pour visualiser les prix d'une céréale par région et par année nous avons utilisé une carte du Sénégal au format GeoJson crée en utilisant des fichier shapefiles sur site <https://mapshaper.org/> pour obtenir une carte détaillée du pays avec les différentes régions.

4. Visualisations et graphes :

Sur le site nous avons créé des line chart à pour chaque céréale une visualisation interactive sur une carte ainsi que des bars chart groupés.

Le chart groupé est visible lorsqu'on clique sur une région de la carte. La région choisie est transmise via une variable de session qui sera utilisé comme paramètre pour le filtrage des données.

Outils et Documentations

SiteWeb :

- Le site est réalisé avec HTML et JS
- Le design CSS a travers un Framework du nom de MaterializeCSS (<https://materializecss.com/>)

Visualisations :

- Avec du Javascript à travers la bibliothèque D3JS (Data-Driven-Document).

Traitement des Données :

- Pour le traitement on a utilisé Pandas de python.

Formats manipulés :

- CSV
- JSON
- SHP, SPX

Webographie :

<https://d3js.org/>

<https://www.d3-graph-gallery.com/>

<https://stackoverflow.com/>

<https://www.w3schools.com/>

<http://learnjsdata.com>

<http://kaggle.com/>

