Date 15 février 2017

RAPPORT PHASE 2

N° enregistrement de l'équipe DatavizPesticides-1086

Site internet http://www.hydroviz.fr **Contact (email)** hydroviz.fr@gmail.com

Logos des organisateurs et partenaires du concours

Démarche du projet HydroViz

La qualité des ressources d'eau douce est un sujet sensible qui nous concerne tous : il s'agit de l'eau que nous consommons tous les jours de diverses manières, que ce soit pour nous hydrater, nous laver, laver les objets du quotidien, ou encore celle servant à irriguer les cultures… Nous avons donc dégagé deux démarches possibles pour informer le grand public comme les spécialistes sur cette problématique :

- Soit de permettre simplement de constater les faits en offrant des outils statistiques faisant la part belle aux évolutions des chiffres associés à la présence des pesticides ;
- **Soit d'accompagner par différents biais l'utilisateur pour qu'il découvre progressivement et par lui-même le sujet et les informations l'intéressant directement.**

Dans le cadre du projet **HydroViz**, nous avons décidé de mettre en place une solution prenant le parti de cette dernière démarche, tout en **répondant à l'ensemble des cinq défis du concours** (listés à l'article 4.4 du règlement du concours) :

- visualisation des **niveaux de contamination** ;
- visualisation de l'**évolution des niveaux de contamination** ;
- visualisation des pesticides par **catégories** ;
- visualisation des pesticides par **fréquence d'apparition** ;
- mise à disposition des **informations sur les caractéristiques** des pesticides de manière interactive.

Nous sommes convaincus qu'**en plaçant l'utilisateur en tant qu'explorateur** – qu'il s'agisse d'un citoyen désirant s'informer sur la qualité des eaux dans sa région, d'une personne informée désirant aller plus loin dans son apprentissage du sujet, voire d'un expert à la recherche de données précises – la problématique prend une dimension dans laquelle cet explorateur peut à terme **devenir acteur**. Pour ce faire, l'outil que nous proposons se doit d'être didactique, intuitif, de proposer des niveaux de lecture variés, adaptés au niveau de connaissance de l'utilisateur, mais également de lui permettre de progresser.

HydroViz a été développé dès les premières heures tant dans la lettre que l'esprit des **valeurs de l'open source** : la grande majorité des logiciels utilisés sur l'ensemble du *workflow* sont libres, le code source est ouvert et public depuis le départ, mais nous avons également poussé l'exercice en publiant régulièrement des nouvelles sur l'avancement du projet sur les réseaux sociaux. Ceci nous a notamment permis de récolter des avis d'utilisateurs d'horizons très différents qui ont enrichi la liste des idées d'évolutions possibles de l'application, car nous partons du principe qu'**un outil permettant d'aborder une telle problématique de santé publique se doit d'être évolutif et construit en concertation avec les citoyens, les pouvoirs publics, et la communauté scientifique**.

L'ensemble du projet **HydroViz** à ce jour a été pensé, conçu, développé, et finalisé de manière collaborative par Julien Paris et Florian Melki. Vous trouverez en fin de rapport plus de détails sur nos compétences et centres d'intérêt multiples.

Notes :

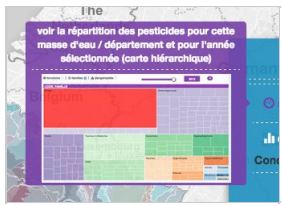
- *Le sommaire se trouve en fin de dossier ;*
- *Le support visuel (*slides*) utilisé lors de la présentation devant jury le 16/02/2017 est consultable en cliquant sur le lien suivant : présentation_Hydroviz ;*
- *Le code source (sous licence GNU GPL) de l'application HydroViz est librement téléchargeable dans son intégralité en cliquant sur le lien suivant : repo GitLab*

Principes de la data visualisation dans HydroViz

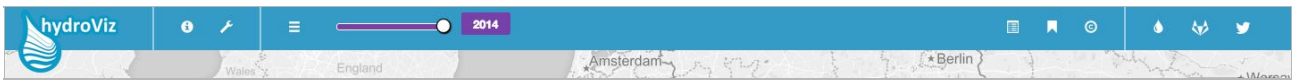
Généralités

HydroViz a été développé dans une approche qui se veut pédagogique et didactique de manière à ce que l'utilisateur ne soit jamais perdu, ni par les fonctionnalités, ni par les contenus. Le but est avant tout de sensibiliser et d'informer l'utilisateur à la problématique de la pollution aux pesticides des eaux souterraines. Ainsi l'utilisateur est d'abord accueilli par un **"pop up" introductif** où est présentée rapidement la problématique accompagnée d'une vidéo de Data Gueule sur l'eau (avec l'accord formel de la maison de production), un présentation des objectifs de l'application, un guide d'utilisation, une présentation des auteurs, et enfin les logos des organisateurs et partenaires du concours.

Des **info-bulles** apparaissent lors du passage de la souris sur l'ensemble des boutons interactifs, expliquant les fonctionnalités à disposition ou guidant l'utilisateur vers les informations complémentaires dont il pourrait avoir besoin.

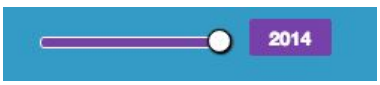


La barre de navigation permet d'accéder à toutes les informations complémentaires relatives au projet HydroViz : guide d'utilisation, *slider* temporel (voir paragraphe suivant), méthodologie, sources et ressources, licences des programmes et des données utilisés, ainsi que les mentions légales, et les contacts.



De ce fait, HydroViz accompagne l'utilisateur pour lui faire prendre en main les trois fonctionnalités principales de l'application que sont : **le *slider* temporel, la carte, et le *treemap*.**

Le *slider* temporel

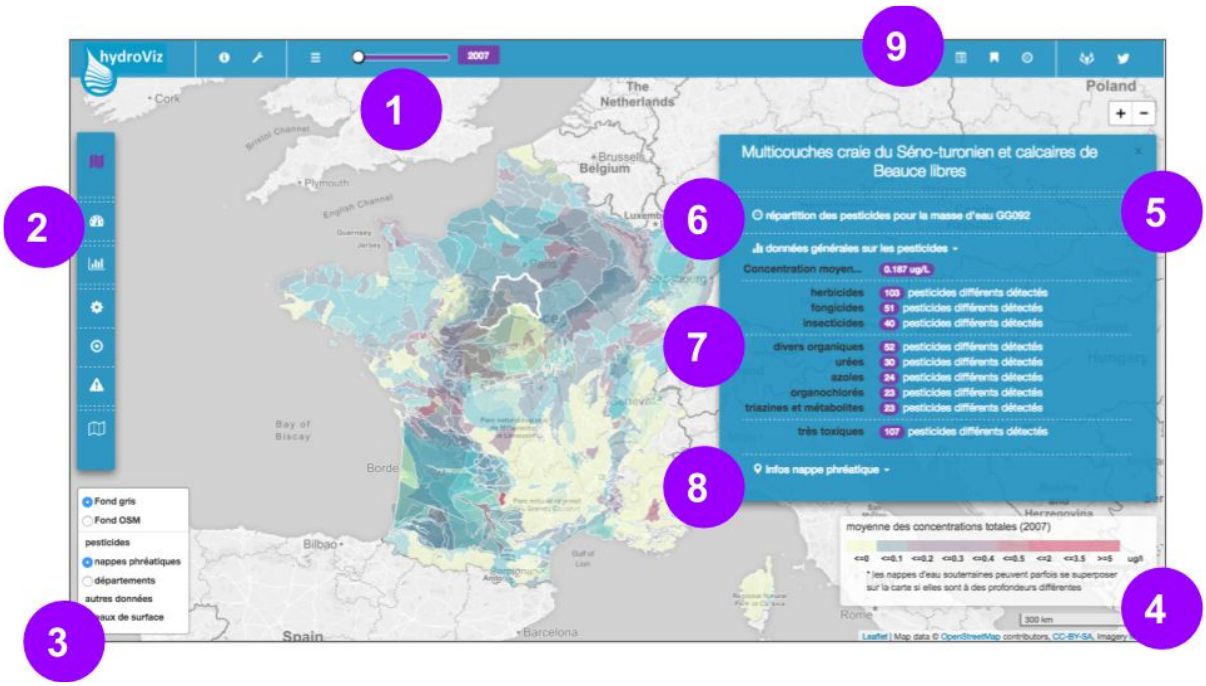


La première dimension sur laquelle l'utilisateur peut jouer est le **facteur temps**. Affichée sous la forme d'un *slider*, cette fonctionnalité permet à l'utilisateur de passer d'une année à une autre quelle que soit la visualisation (carte ou *treemap*). Bien que techniquement réalisable, nous n'avons pour l'instant pas cherché à visualiser simultanément les valeurs des pesticides cumulées sur plusieurs années, car nous désirons que l'utilisateur puisse explorer les données par lui-même, surtout qu'il puisse comprendre intuitivement l'évolution des niveaux de pollution aux pesticides, et qu'il se fasse un représentation concrète de cette évolution.

La carte interactive

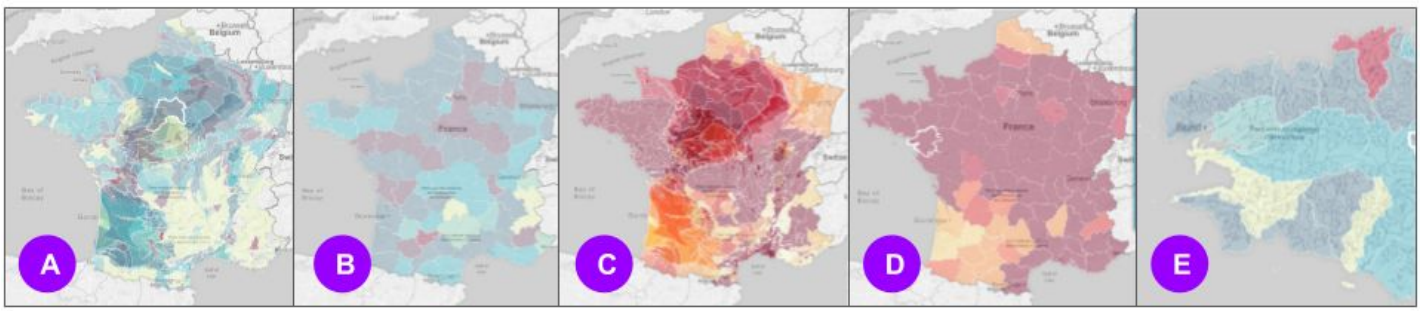
L'utilisateur est placé dans son propre contexte géo-écologique : il connaît peu ou prou sa région, les rivières et lacs qui l'entourent, quel goût à l'eau qu'il boit tous les jours. Utiliser la cartographie comme premier outil de data visualisation permet à tout un chacun de **pouvoir se repérer** rapidement et **donner une réalité plus concrète aux données chiffrées**.

La visualisation de façon cartographique dans HydroViz permet de visualiser la présence de pesticides par concentrations moyennes totales mesurées ou selon une catégorie (fonction, famille, dangerosité), sur le découpage géographique de son choix (département ou nappe phréatique), et sur une année donnée.



Fonctionnalités de la cartographie

1. *Slider* temporel
2. Filtres / toute la France (voir légende suivante)
3. Contrôle des calques (voir légende suivante)
4. Echelles (couleurs et distances)
5. Fenêtre informative pour la zone sélectionnée
6. Lien vers le treemap pour la zone sélectionnée
7. Données de base sur les pesticides pour la zone sélectionnée
8. Informations sur la nappe d'eau sélectionnée
9. Onglets complémentaires dans la barre de navigation



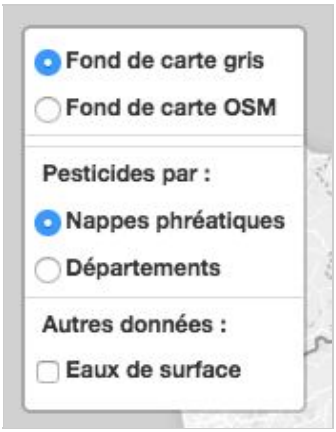
Calques de l'outil cartographique

A. Concentrations moyennes totales / par nappes phréatiques
B. Concentrations moyennes totales / par départements
C. Dénombrement des pesticides / par nappes phréatiques et par catégorie
D. Dénombrement des pesticides / par départements et par catégorie
E. Affichage (optionnel) du fond de carte des eaux de surface

Nous avons enrichi par nous-mêmes la base de données fournie avec les **données de dangerosité** en nous basant - pour le moment - sur un rapport de l'organisation mondiale de la Santé (WHO) recoupé par des informations tirées de la base de données TOXNET (faute d'avoir trouvé des bases de données propres à l'Europe facilement exploitables). Il nous semblait important de communiquer ce type d'information au grand public non spécialiste.

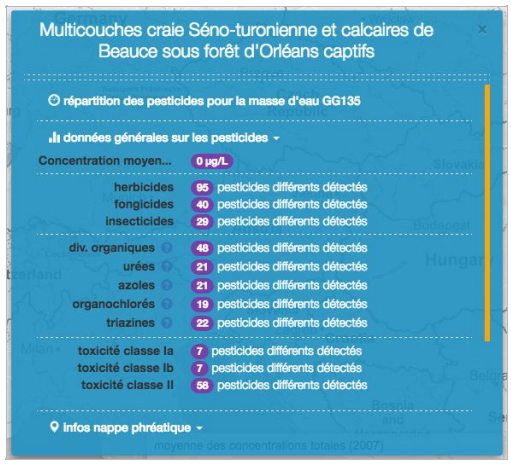
Lorsqu'une des catégories ou que le bouton "concentrations moyennes totales" est sélectionné (cf. **6** ou **2**), la légende (cf. **4**) correspondante apparaît. Le code-couleur de la légende (le rouge synonyme d'un plus grand risque, une couleur pastel de "sécurité") permet à l'utilisateur de focaliser rapidement sur les informations qu'il juge pertinentes.

HydroViz offre la possibilité à l'utilisateur d'afficher ou d'alterner entre différents calques (cf. **3** : fenêtre de contrôle en bas à gauche) permettant de varier les fonds de carte ("fond gris" et "fond OSM") et surtout les entités géographiques de représentation des pesticides :



- Affichage **par département** (cf. **B** et **D**) : c'est le niveau le plus intuitif, car il renvoie à un élément géographique commun.
- Affichage **par nappe phréatique** (cf. **A** et **C**) : ce niveau est plus avancé que le précédent. S'il est moins évident de s'y repérer étant donné que les nappes peuvent se superposer, sa précision demeure meilleure.
- L'affichage des **eaux de surface** (cf. **E**) est également proposé en option. Bien que nous ne possédions pas encore les données relatives à la présence des pesticides dans celles-ci, nous avons l'intuition que ces données peuvent intéresser, ne serait-ce que pour se situer, ou parce que les nappes phréatiques et les eaux de surface peuvent parfois communiquer. La visualisation des données de contamination aux pesticides des eaux de surface serait un des chantiers futurs du développement d'HydroViz (voir chapitre "évolutions possibles" plus bas).

Enfin, un clic de l'utilisateur sur la zone (département ou nappe phréatique) de son choix permet d'afficher une boîte d'information (cf; **5**) contenant des détails sur les métadonnées relatives à l'entité géographique choisie, et d'accéder à la vue « avancée » *treemap* (cf. **6** et section suivante).



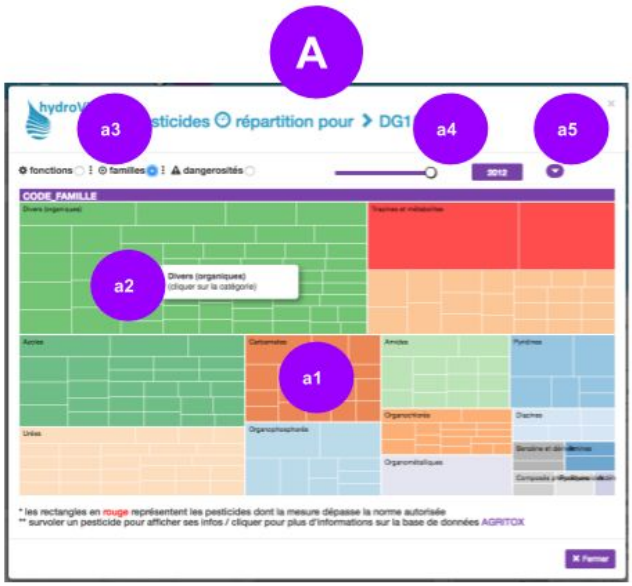
Bien que nous ayions généré un jeu de données géolocalisées pour les stations (cf fichier `stations_web_carto_topo.json` dans le répertoire `.app/static/data/carto_web`) nous avons fait le choix éditorial de ne pas inclure de représentation des stations de mesure sur la carte : en effet dans une approche qui s'adresse en priorité au grand public nous avons jugé cette information redondante avec la datavisualisation des données par nappe d'eau. Cela n'aurait pas apporté beaucoup d'informations supplémentaires, et poserait également des

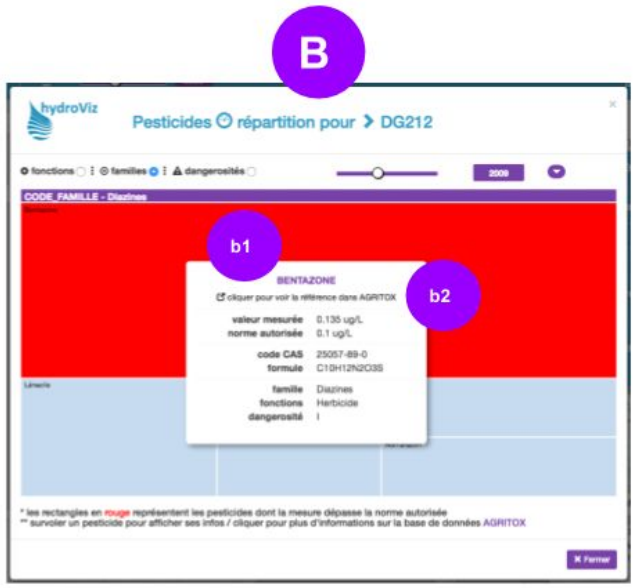
questions de sécurité (une simple précaution dans un scénario où des personnes ou des entités mal intentionnées voudraient artificiellement corrompre les données ou s'attaquer au matériel / rendre la station inutilisable). La probabilité d'un tel scénario serait peut-être à évaluer avant de proposer une telle fonctionnalité.

- ## Le treemap

Le *treemap* est une visualisation hiérarchique permettant de mettre en évidence, dans un ou plusieurs groupes, les éléments prédominants. Dans notre cas, il s'agit donc de **représenter les différents pesticides, regroupés par catégorie** (fonction, famille, et dangerosité), **et par année**.

Chaque catégorie est "zoomable" via un clic afin d'accéder en détail aux caractéristiques de chaque pesticide. Un passage de la souris sur un pesticide affiche une bulle d'information contextuelle donnant les détails du pesticide (valeur mesurée, norme autorisée, fonction,...) et permet par un autre clic d'ouvrir sa fiche détaillée sur le site AGRITOX de l'ANSES s'il y est répertorié.



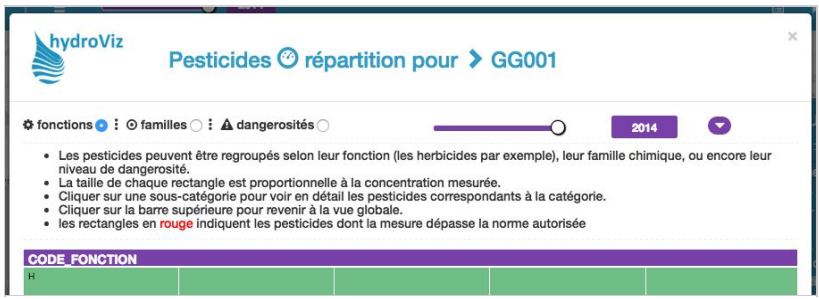


A - répartition globale de pesticides pour une catégorie donnée

a1 - treemap des sous-catégories
a2 - aide contextuelle
a3 - sélection de la catégorie à afficher
a4 - slider temporel
a5 - rubrique d'aide

B - "zoom" sur une sous-catégorie

b1 - caractéristiques du pesticide survolé
b2 - lien (le cas échéant) à la page du pesticide survolé sur le site AGRITOX de l'ANSE



Une rubrique d'aide (cf. **a5**) peut également être dépliée, donnant plus d'indications sur l'utilisation de l'outil *treemap*.

Afin de répondre à l'ensemble des contraintes du concours nous avons dû procéder à certains choix dans l'implémentation de notre solution de data visualisation. Ces choix sont expliqués en détail dans les chapitres suivants ("Traitement des données et algorithmes", "Licences des logiciels utilisés", "Documentation technique"), toutefois les raisons suivantes nous ont servi de fil rouge durant toute la période de développement :

● Conserver une ligne éditoriale

L'application HydroViz a été conçue autour d'une ligne éditoriale claire : **s'adresser en priorité au grand public, en proposant des outils les plus intuitifs et didactiques possibles**, pour qu'il puisse avoir une vue d'ensemble de cette problématique complexe qu'est la pollution aux pesticides. Nous avons tenté de garder en tête cet aspect durant tout le développement, mais nous avons également fait un effort de communication (sur Twitter, Facebook, en envoyant des liens à des proches spécialistes ou non...) afin d'avoir des retours d'expérience sur tous les aspects de la solution que nous avons proposée et l'améliorer en continu.

Même si le code permettant à l'application de fonctionner est relativement élaboré pour qui n'est pas développeur et que les données possèdent en elles mêmes un certain degré de complexité, nous avons tout de même tenté de rendre l'interface (UI-UX) la plus dépouillée et épurée possible afin que l'outil puisse être pris en main par tout un chacun.

● Contraintes de temps

Afin de tenir les délais impartis nous avons concentré les efforts de développement sur les fonctionnalités que nous jugions centrales (*slider*, cartographie, *treemap*) et un certain nombre de "sous-fonctionnalités" (liens vers les bases de données telles que AGRITOX ou SANDRE, info-bulles...). L'idée était de proposer pour la date du premier rendu **une application minimaliste mais prête à l'emploi** - une sorte de "proof of concept" - qui pourrait être ensuite améliorée ou enrichie de nouvelles fonctionnalités ou de nouveaux jeux de données après concertation avec les publics et des experts.

Nous avons listés plusieurs problématiques et fonctionnalités qui nous semblaient intéressantes d'approfondir, dans la perspective de la prolongation du projet HydroViz. Ces pistes de travail sont développées plus bas dans le chapitre "Évolutions possibles" : elles partent de l'idée de rendre un tel outil plus participatif, afin de le muer petit à petit en réel outil d'aide à la prise de décision pour les collectivités, ou/et en un outil permettant une meilleure traçabilité des pesticides dans les eaux souterraines ou de surface, afin qu'il devienne à terme un outil complet permettant de médiatiser cette problématique importante de santé publique.

● Contraintes liées aux questions de propriété intellectuelle

Les deux membres de l'équipe HydroViz possédant tous deux une culture et une pratique des logiciels libres ou de l'open source les choix d'implémentations techniques sont venus assez naturellement : Python, JavaScript, D3, Leaflet, étaient des outils dont nous avions un certain degré de connaissance et de maîtrise.

Nous possédons également tous deux **une sensibilité et des convictions d'ordre plus éthique sur l'ouverture des données et des codes sources**, convictions que nous traduisons chacun dans des activités différentes : dans un engagement associatif pour l'ouverture des données, dans la pratique du Do It Yourself, ou par notre implication dans des projets ouverts et citoyens par exemple.

● Un projet en développement continu

HydroViz est au final un projet qui a été **conçu et développé de manière la plus ouverte et transparente possible**, en cohérence avec nos convictions et nos compétences, tout en respectant l'ensemble des contraintes du concours. Il vise à demeurer un projet en constante évolution, ouvert aux critiques et aux propositions, et nous espérons qu'il s'enrichira au fil des demandes et des besoins.

Traitement des données et algorithmes

Nous avons utilisé la totalité des données fournies dans le cadre du concours (fonds de carte et statistiques), auxquelles nous avons pris la liberté d'ajouter des données relatives à la dangerosité des pesticides (classées par niveaux de toxicité, source WHO). A l'aune de nos connaissances actuelles et dans le cadre d'une approche orientée grand public nous avons choisi de ne pas faire appel à davantage de modèles physiques ou autres (de diffusion de pesticides, d'interaction avec les eaux de surface ou avec l'air), sinon de nous consacrer à la pleine exploration des données à disposition.

En étape de prétraitement ainsi qu'en utilisation "live", les données sont traitées et analysées avec la bibliothèque open source Python Pandas.

En amont les jeux de données originaux ont d'abord été formatés et intégrés dans des jeux spécifiques (objets Pandas). La technologie offerte par Python et la bibliothèque Pandas permettent de faire des opérations statistiques très rapidement sans altérer la base de données, ou encore de regrouper les informations dans des tables dédiées. Les données cartographiques d'origine ont quant à elles été optimisées en amont pour une utilisation web (GeoPandas et TopoJSON).

Enfin, les systèmes de requêtes client (IO) ont été pensés pour renvoyer au client le plus rapidement possible des données les plus légères et les plus synthétiques, optimisant le système pour une utilisation web. Les données les plus volumineuses (fond de carte des nappes phréatiques notamment) sont chargées en une seule fois en arrière-plan à l'ouverture de l'application. Ce chargement initial est "dissimulé" par l'ouverture d'une fenêtre popup d'introduction, rendant l'impression d'une navigation plus fluide.

De manière générale le choix a été fait de porter l'accent sur le facteur temps afin de constater dynamiquement mais aussi visuellement l'évolution de la pollution.

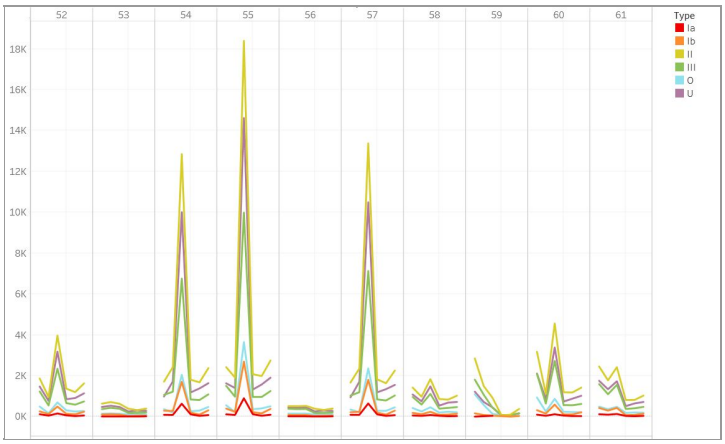
● Data mining / exploration préalable des données:

Dans un premier temps, nous avons parcouru les jeux de données statistiques avec le logiciel Tableau, et les données géolocalisées avec GeoPandas, avec le logiciel de SIG open source QGiS, ainsi que MapShaper. Nous avons parcouru les données sans a priori, sans idée préconçue de ce que nous cherchions. Fidèles à notre démarche, notre but était alors de découvrir et de nous approprier les données dans une posture de néophytes. Cette étape nous a alors permis de tester plusieurs options de data visualisation et de nous familiariser avec les lexiques propres aux données de l'ADES/SANDRE. Peu à peu nous avons ainsi pu nous projeter dans les dimensions du quotidien, plus familières : le temps et l'espace.

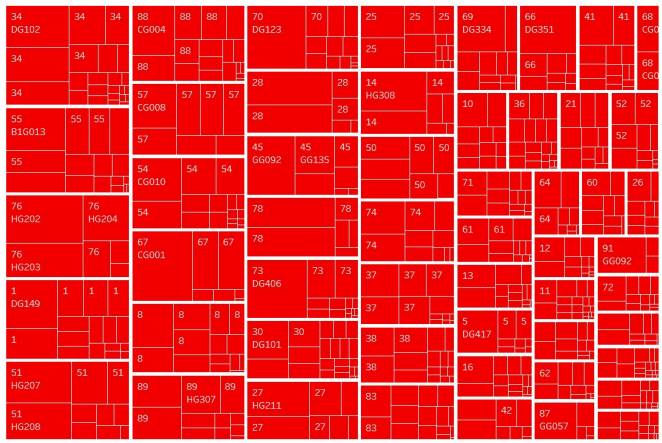
Le temps, car il symbolise le changement et permet de rendre compte d'une évolution, celle de la présence des pesticides. L'espace, car il est celui dans lequel nous naviguons. Il nous est rapidement apparu que le concept de "station", aussi précis qu'il soit, risquait de ne pas "parler" au grand public. En revanche, la notion de masse d'eau, de ville ou de département semblaient être des notions plus communes et intelligibles, et rendaient mieux l'idée d'un continuum et d'une diffusion des polluants sur le territoire.

Si nous avons pu constater des tendances, des endroits géographiques davantage touchés que d'autres (la région de la Lorraine, ou le département de l'Hérault en première ligne), nous nous sommes en revanche trouvés incapables de pouvoir l'expliquer (n'étant ni l'un ni l'autre hydrologues). Ceci explique notre choix de seulement exposer/représenter les données (mesures de concentrations ou dénombrement des pesticides) au lieu de risquer de provoquer la confusion chez l'utilisateur, en lui proposant soit des graphiques sans pouvoir y adjoindre une explication plausible, soit des hypothèses explicatives sans base scientifique réelle.

Voici quelques extraits des explorations préalables des données que nous avons effectuées avant de réellement commencer à concevoir et développer l'application HydroViz.

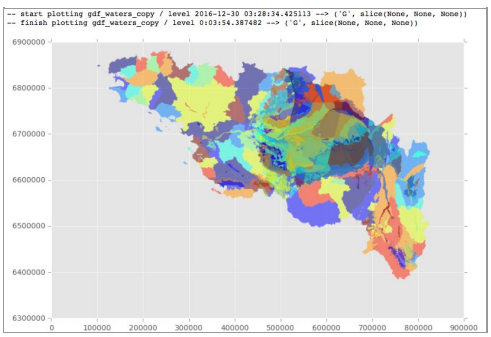


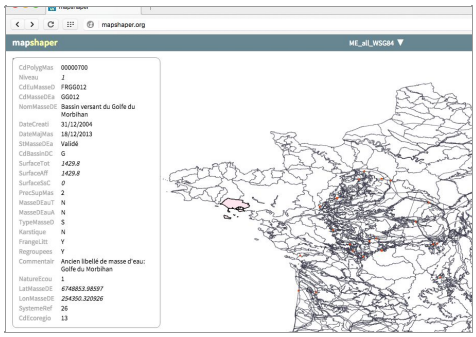
En abscisse, nombre d'éléments présents par type de dangerosité; en ordonnée, double axe département / temps.
Les pics ont lieu en 2009, mais il nous est impossible d'en identifier la cause.

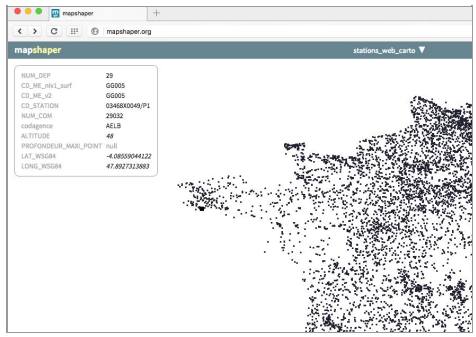


Treemap représentant la répartition hiérarchique de pesticide de toxicité "Ia" (les plus dangereux) par département

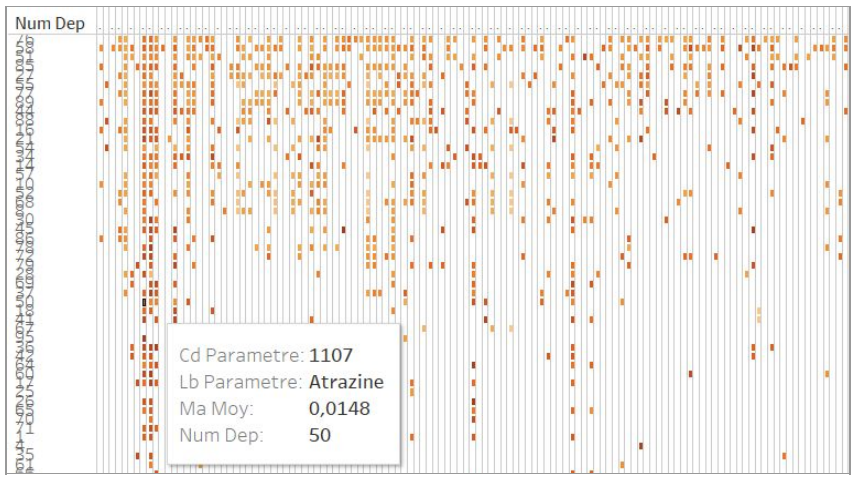
---







Exploration préalable sur GeoPandas et MapShaper des fonds de carte et des données géolocalisées (masses d'eau et stations).



Présence des pesticides par département, classés par Code Paramètre.
Un code couleur plus sombre indique une forte concentration moyenne.

Les principales opérations appliquées aux jeux de données originaux sont les suivantes :

- ## Prétraitement et de formatage préalable des données :

Au vu de quelques soucis techniques et de certains échanges sur le forum du concours concernant les jeux de données source au format .csv, nous avons fait le choix d'utiliser comme matériel source uniquement les fichiers .xlsx (de type `ma_qp_fm_rcsrco_pesteso_XXXX.xlsx` et `moy_tot_quantif_XXXX.xlsx`) pour les données de mesure de concentrations. Les fichiers sources originaux sont stockés à part dans le répertoire `.app/static/data/stats`.

Les différents tableaux .xlsx de mesures de pesticides (concentrations moyennes totales et concentrations pesticides par pesticides) sont nettoyés, copiés, puis fusionnés afin de produire plusieurs fichiers .csv (plus rapides à lire par Pandas), dont deux tables principales, qui seront ensuite consommées par l'application. Les fichiers destinés à une consommation web sont stockés à part dans le répertoire `.app/static/data/stats_web.` Les deux tables principales sont :

- une table des mesures des pesticides et concentrations moyennes totales (lignes) par code masse d'eau (colonnes) ;
- une table des mesures des pesticides et concentrations moyennes totales (lignes) par départements (colonnes).

Données statistiques

Les opérations expliquées dans le présent paragraphe se réfèrent aux scripts du fichier `pesticides_analysis_03.ipynb`, servant à compiler, nettoyer, préparer les données originelles en vue de leur consommation par l'application Hydroviz.

Les différents jeux de tableaux de mesures et les informations sur les stations de mesure sont d'abord lus et regroupés en plusieurs objets Pandas. Les informations de toxicité sont alors intégrées, et les masses d'eau sont indexées en fonction des stations et départements qui leur correspondent.

Les données affichées pour les masses d'eau souterraines sont obtenues en faisant la moyenne des mesures des stations qui possèdent des données pour la masse d'eau considérée. A cette fin le script de préparation des données aboutit à l'écriture du fichier `AV_ME_web.csv` dans le répertoire `./app/static/data/stats_web`.

Le jeu complémentaire `AV_dpt_web.csv` généré et visant à afficher les données selon les départements procède du même principe que précédemment : les données affichées pour les départements sont obtenues en faisant la moyenne des mesures des stations situées dans le département considéré.

Données géographiques (fonds de carte)

Les masses d'eau possédant exactement les mêmes codes ("CdMasseDEa") ont été regroupées dans des mêmes entités géométriques pour optimiser l'affichage. Les fichiers shapefile ont été transformés en fichiers TopoJSON (format remarquable créé par M. Bostock, créateur de D3.js), leur géométrie a été simplifiée pour un gain de poids considérable, mais cela tout en conservant la topologie de l'ensemble. Une projection mercator a été calculée sur les fichiers TopoJSON pour une utilisation dans Leaflet (crs_WSG84). Ces transformations ont étés faite avec GeoPandas et MapShaper (logiciels tous deux ouverts).

- ## Intégration de bases de données supplémentaires

Dangerosité des produits

Nous nous sommes basés des informations issues du site de l'Organisation mondiale de la santé, recoupées par des informations trouvées sur le site TOXNET, pour retrouver le niveau de dangerosité des pesticides. Nous avons - pour le moment - fait ce choix faute de trouver une base de données européenne dans un format convenable pour l'analyse. La base de donnée que nous avons donc généré pourra être très simplement changée si nécessaire, cette opération ne nécessitant que de mettre à jour les tables grâce au script Jupyter

pesticides_analysis_03.ipynb : voir plus bas le chapitre “documentation technique”, § ”Préparation des données / intégration de nouveaux jeux de données”.

Eaux de surface

Le fond de carte des eaux de surface provient de la plateforme officielle datagouv.fr. Nous avons choisi d'inclure cette base dans la perspective d'une intégration future dans le même outil HydroViz des données de pollution aquatique, qu'elle soit de surface ou souterraine.

- **Sélection et analyse des données**

Requêtes côté client

Les requêtes se font de manière asynchrones entre le serveur et le client via SocketIO. Cette solution permet de garder en cache les fonds de carte et un certain nombre de paramètres dans le script javascript pour les requêtes et l'affichage des cartes et des tableaux.

Analyse côté serveur

Les algorithmes de sélection et d’analyse des données sont écrits en Python dans le répertoire app/scripts. Les données (tableaux .csv dans le répertoire app/static/data/stats_web) sont préchargées une seule fois lors de l'initialisation du serveur (local ou cloud) et gardées en mémoire sous forme d'objets Pandas non altérables (se référer à l’ensemble des scripts du fichier load_data.py).

Des fonctions généralistes ont été écrites afin de bénéficier de toute la puissance de Pandas de manière dynamique (fichier get_data.py). En fonction des requêtes envoyées par le client ces scripts permettent d’opérer des “slices” dans les données brutes (sélection par pesticides, par année, …) de manière non-spécifique à partir d’un seul objet-sélection : une seule et même “classe” Python nommée GetDataSlice.

Les scripts écrits (donc l’objet-sélection Python) du fichier get_data.py sont appelés depuis le fichier views.py : depuis le fichier views.py les objets Pandas sont analysés (dénombrement ou min/max par exemple), puis transformés en objets JSON, et finalement envoyés de manière asynchrone via SocketIO au client afin d’être transformés/visualisés par D3 ou Leaflet. Les données consommées par la cartographie ou par les treemaps proviennent de deux fonctions différentes écrites dans views.py (def send_AV_slice() pour la carto, et def send_AV_tree() pour le treemap), mais c’est toujours la même “classe” GetDataSlice qui est appelée au final dans get_data.py.

Les opérations de dénombrement des pesticides ignorent les valeurs inexistantes (valeurs telles que “blank”, None”, Nan”) ainsi que les valeurs nulles (strictement égales à 0.0). Se référer à la fonction Python/Pandas AV_counts_by_func_fam_type() , ligne 195, dans le fichier get_data.py (situé dans le répertoire ./app/scripts)

Les données sont consommées côté client “en l’état” par les scripts Javascript (Leaflet et D3), évitant ainsi tout calcul agrégatif supplémentaire.

Représentation et échelles

Les pesticides dont la concentration donne une valeur nulle ou est inexistante sont automatiquement ignorés dans le *treemap* (calculant alors une aire du rectangle nulle). Se référer à la fonction globale Javascript/D3 makeTreeMap() , lignes 416-661, dans le fichier treemap_c.html (situé dans le répertoire ./app/templates)

Le choix a été fait d’utiliser deux échelles logarithmiques statiques : l’une pour les mesures de concentrations (nuances allant du jaune clair, au bleu clair, jusqu’au pourpre) ; l’autre pour les mesures de dénombrement de pesticides (nuances allant du jaune clair au rouge foncé). Le choix de ne pas utiliser une délimitation par quantiles mais des échelles fixes (min et max restant toujours les mêmes) permet une meilleure comparaison entre les mesures d’une année sur l’autre pour l’utilisateur.

Le choix d'utiliser des échelles logarithmiques permet de "caler" des couleurs sur des valeurs clés (0,1 µg/L pour les concentrations, 50 pesticides différents détectés pour le dénombrement par exemple) tout en conservant des valeurs maximales parfois très largement supérieures.

Se référer aux fonctions Javascript/Leaflet `refreshMap()`, `refreshTopoLayer()`, `refreshLegend()`, `handleLayer()`, `chooseColorScale()`, dans le fichier `map.html` (situé dans le répertoire `./app/templates`) ; ainsi qu'aux variables min/max et couleurs dans le fichier de configuration de l'application `app_settings.py` (situé dans le répertoire `./app/scripts`)

Cette architecture permet de minimiser la quantité de code écrit tout en rendant les scripts les plus polyvalents possible (non-spécifiques), et de respecter l'intégrité des données initiales. Ainsi l'application est plus "élégante" et plus facilement adaptable à de nouveaux jeux de données et à l'implémentation de nouvelles fonctionnalités.

La totalité du code est open source et consultable à l'adresse suivante :

https://gitlab.com/Julien_P/concours_pesticides

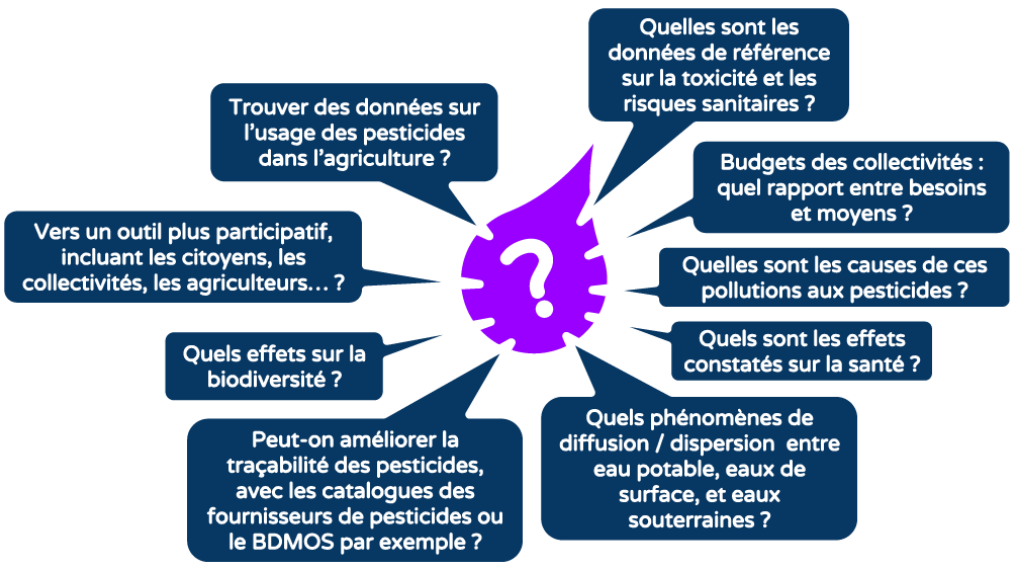
Evolutions possibles

En l'espace d'environ 5 semaines de travail intensif nous avons exploré, nettoyé et enrichi les données, conçu l'architecture et le design d'une application évolutive et ouverte, intégré les principales fonctionnalités, et finalement documenté et mis en ligne un site fonctionnel : http://www.hydroviz.fr.

Toutefois nous avons dû faire des choix dans l'implémentation des fonctionnalités d'HydroViz. L'application a été développée afin d'être aisément évolutive. Ainsi, même si nous n'avons pu les implémenter, nous gardons en tête la **possibilité d'intégrer à l'avenir les fonctionnalités et les améliorations suivantes**, lesquelles sont pour la plupart issues de retours d'utilisateurs (proches, journalistes, chercheurs, …) :

- **Cartographie des concentrations de pesticides pris un à un** (par concentration mesurée ou delta par rapport à la norme) : pour l'instant les niveaux de couleurs affichés sur la carte correspondent soit à la concentration moyenne totale mesurée ("MOYPTOT"), soit au nombre de pesticides différents trouvés pour une sous-catégorie donnée ("famille/Azoles" par exemple). Afficher les niveaux de concentrations pour chacun des pesticides pris un à un permettrait des recherches plus fines encore.
- **Mise à jour des données de toxicité** avec une base de données propre à la réglementation européenne.
- **Intégration des budgets des collectivités et institutions ayant des missions relatives à la qualité des eaux** : ces données permettraient d'évaluer plus finement le rapport entre les besoins (concentrations de pesticides) et les moyens mis en oeuvre par les pouvoirs publics pour assurer la mise aux normes et la non-dangerosité des réserves d'eau douce.
- **Outils visant une approche participative :** forum, formulaires, permettant de faire "remonter" des informations et des données de manière décentralisée
- **Concentrations de pesticides dans les eaux de surface** : à la condition d'obtenir les données correspondantes nous serions en mesure de les intégrer dans HydroViz relativement rapidement.
- **Un outil de sélection des nappes par niveau de profondeur**.
- **Données d'utilisation des pesticides dans le secteur agricole** : de la même manière nous avons trouvé des fonds de carte concernant les parcelles agricoles (registre agricole). Sii nous avions à disposition des données sur la consommation des pesticides nous serions à la fois en mesure de visualiser / mettre en perspective les niveaux de contamination avec l'utilisation de ces pesticides.
- **Données sur les fournisseurs de pesticides :** dans le prolongement du paragraphe précédent il serait intéressant d'avoir les catalogues des fournisseurs afin d'améliorer la traçabilité des produits détectés.

Un onglet intitulé "vos retours" dans la barre de navigation synthétise ces idées issues de notre *crowdsourcing*.



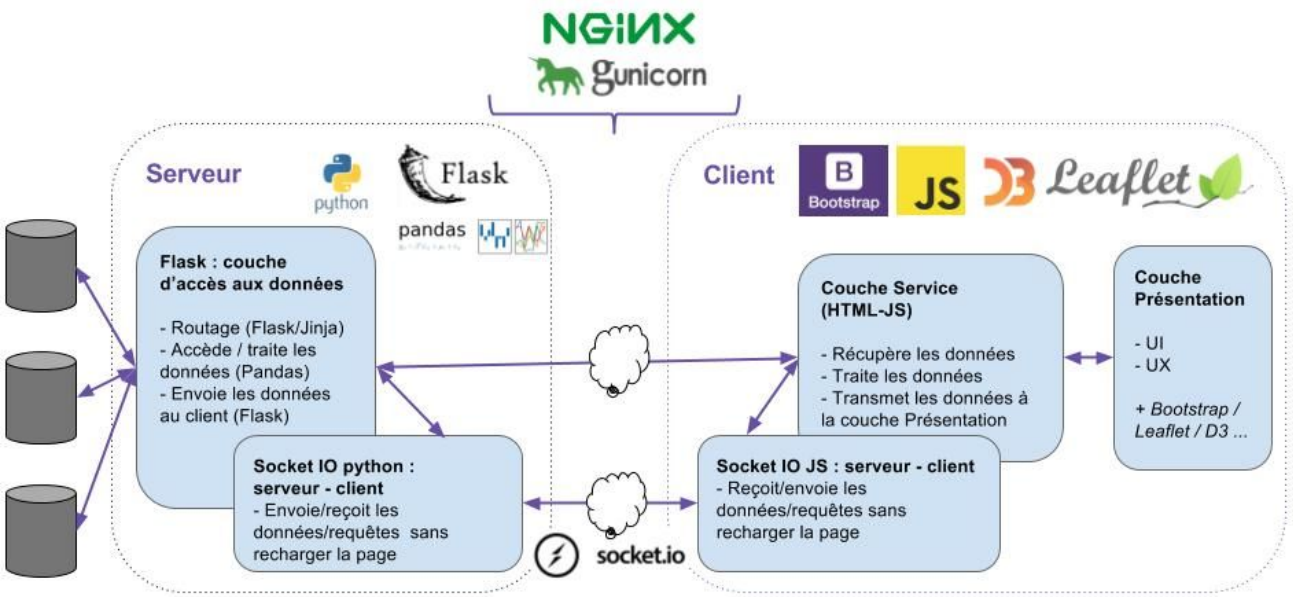
Licences des logiciels utilisés

HydroViz est entièrement développé en open source.
L'application est publiée sous licence GNU GPL sur la plateforme GitLab.

Nous avons fait le choix d'une licence ouverte GNU GPL car elle nous paraissait à la fois plus permissive, plus simple, et moins contraignante que d'autres telles que les licences MIT ou creative commons en termes de réutilisation du code à des fins commerciales ou non. Ce choix s'explique également par le fait qu'une licence GNU GPL nous semblait plus à même de de respecter sans ambiguïté et globalement l'ensemble des obligations légales des licences des logiciels utilisés (listées ci-après).

Architecture de l'application

L'application est principalement écrite en Python et en JavaScript : le serveur est développé avec Flask, les requêtes client-serveur avec SocketIO / flask-SocketIO, l'analyse et le nettoyage de données avec les bibliothèques Python Pandas et GeoPandas.



- Python : langage de programmation open source (OSI-approved open source license)
- Flask : microframework en Python sous licence BSD créé par Armin Ronacher et alii.
- Pandas / GeoPandas / Numpy : bibliothèques Python pour l'analyse de données et le calcul scientifique (licence BSD)
- JavaScript (et JQuery) : langages de programmation
- SocketIO / flask-socketIO : requêtes asynchrones serveur-client (en Python et javascript, licence MIT)

Design de l'application

Les outils de data visualisation sont principalement : Bootstrap, D3.js, et Leaflet. Le design du logo d'HydroViz a été fait avec le logiciel libre InkScape.

- **Bootstrap** : framework client open source (licence MIT)
- **D3.js** (Data Driven Documents) : bibliothèque de data visualisation (Copyright 2010-2016 Mike Bostock all rights reserved, avec autorisation de modifications et de redistribution)
- **Leaflet** : bibliothèque de cartographie en javascript (Copyright (c) 2010-2016, Vladimir Agafonkin / Copyright (c) 2010-2011, CloudMade All rights reserved, avec autorisation de modifications et de redistribution)
- **FontAwesome** : bibliothèque d'icônes (licence MIT)
- **Inkscape** : Inkscape est un logiciel libre de dessin vectoriel sous licence GNU GPL. Il gère des fichiers conformes avec les standards XML, SVG et CSS du W3C.

Bases de données (sources)

Les données servant à la datavisualisation ont comme sources principales : Etalab (portail open data national), Open Street Map. Les données ont été préparées en amont pour une consommation web avec les bibliothèques Python Pandas, geoPandas citées plus haut en utilisant l'outil Jupyter lors de la préparation des scripts

- **Etalab** : portail opendata
- **Open Street Map** : fonds de cartes (license ODbL)
- **Jupyter** (optionnel) : outil notebook Python (Copyright (c) 2015, Project Jupyter, avec autorisation de modifications et de redistribution)

Code source et hébergement

Le code source est en open source sur la plateforme GitLab. L'application est hébergée sur Digital Ocean. Les services de proxy et web utilisés sont Nginx et Gunicorn.

- **GitLab** : plateforme libre de partage du code source (licence MIT)
- **Digital Ocean** : hébergement cloud sur serveur Ubuntu Ubuntu 14.04.5 x64 (2 CPU / 4 Go)
- **Nginx** : logiciel libre de serveur Web (ou HTTP) ainsi qu'un proxy inverse (Copyright (C) 2002-2017 Igor Sysoev, Copyright (C) 2011-2017 Nginx, Inc. All rights reserved, avec autorisation de modifications et de redistribution)
- **Gunicorn** : logiciel libre serveur Python WSGI HTTP pour UNIX (2009-2016 (c) Benoît Chesneau / benoitc@e-engura.org 2009-2015 (c) Paul J. Davis / paul.joseph.davis@gmail.com)

Documentation technique *

** Se reporter au fichier README en ligne sur le repo GitLab (https://gitlab.com/Julien_P/concours_pesticides)*

Pré-requis techniques

- **Bibliothèques globales :**
  - Python 2.7
  - Librairies Python : Pandas, geoPandas, flask-socketio
- **Traitement et nettoyage préalable des données :**
  - Jupyter (optionnel)
- **Installation sur serveur :**
  - NGINX
  - Gunicorn (Python)
  - Eventlet (Python)
  - Configuration serveur utilisé : ubuntu 14.04 x64 | 4 Go RAM minimum

---

Préparation des données / intégration de nouveaux jeux de données

- Ajouter les nouveaux jeux de données (fichiers .xls) dans le répertoire *./statics/data/stats*
- Lancer le script `pesticides_analysis_03.ipynb` avec *Jupyter*
- Dans le script `pesticides_analysis_03.ipynb` :
  - Changer la variable `copies_done` de `True` à `False` ;
  - Ajouter la nouvelle année à la liste `years` ;
  - Ajouter le nom du fichier .xls dans les listes correspondantes ;
- Lancer le script (qui updates les fichiers dans le répertoire `./app/static/data/stats_web`)
- relancer *gunicorn* :

```
$ pkill gunicorn
$ gunicorn --bind 0.0.0.0:5000 —-timeout=120 --workers=1 —-worker-class eventlet wsgi:app &
```

Utilisation locale

- Cloner le projet *hydroviz* depuis gitlab :

```
$ git clone git@gitlab.com:Julien_P/concours_pesticides.git
```

- installer, créer et activer un environnement virtuel :

```
$ pip install virtualenv
$ sudo virtualenv venv
$ source venv/bin/activate
```

- Installer les bibliothèques Python (Flask, pandas, etc...) dans l'environnement virtuel :

```
(venv)$ pip install -r requirements.txt
```

- Lancer hydroviz en "debugging mode" :

```
(venv)$ python run_pesticides.py
```

- Dans le navigateur ouvrir l'adresse :

```
http://127.0.0.1:3000
```

Serveur physique (cloud)

- Mettre à jour ubuntu :

```
$ sudo apt-get update
```

- Installer GIT sur le serveur :

```
$ sudo apt-get install git
```

- Cloner le projet hydroviz depuis gitlab :

```
$ mkdir apps
$ cd app
$ git config --list
$ git init
$ git clone git@gitlab.com:Julien_P/concours_pesticides.git
```

- configurer le SSH de votre serveur...
- Configurer le *firewall* du serveur pour socketIO (port 5000), NGINX/Gunicorn (port 8000, www) :

```
$ sudo ufw allow www
$ sudo ufw allow 8000
$ sudo ufw allow 5000
$ sudo ufw enable
$ sudo apt-get update
$ sudo apt-get install ntp
```

- Installer NGINX sur le serveur :

```
$ sudo apt-get install nginx
$ service nginx restart
```

- Installer Python, PIP, et les bibliothèques :

```
$ sudo apt-get install python-pip python-dev
$ pip install -r requirements.txt
$ pip install gunicorn
$ pip install eventlet
```

- Configurer NGINX (reroutage du port 5000 à la racine) :

```
$ cd ~/etc/nginx/sites-enabled`
```

- Créer le fichier de configuration NGINX pour hydroviz

```
$ sudo vi hydroviz
ESC + i
```

- copier/coller dans le fichier de configuration NGINX les *settings* suivants :

```
# configuration containing list of application servers
upstream app_server {
  server 0.0.0.0:5000 fail_timeout=0;
}
# configuration for Nginx
```

```
server {
 # running port
 listen 80 default_server ;
 server_name yourdomain.com ;
 # Proxy connection to the application servers
 location / {
   proxy_pass http://app_server ;
   proxy_redirect off ;
   proxy_set_header Host $http_host;
   proxy_set_header X-Real-IP $remote_addr;
   proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
   proxy_set_header X-Forwarded-Host $server_name;
 }
}
```

- Enregistrer le fichier de configuration NGINX pour hydroviz :

  ```
  ESC + :wq + ENTER
  ```

- Tester les erreurs de syntaxe en tapant :

  ```
  $ sudo nginx -t
  ```

- Redémarrer NGINX afin de prendre en compte la nouvelle configuration :

  ```
  $ sudo service nginx restart
  ```

- Lancer l'application : aller au même niveau que le fichier wsgi.py et démarrer l'application avec Gunicorn :

  ```
  $ cd apps/concours_pesticides
  $ gunicorn --bind 0.0.0.0:5000 —-timeout=120 --workers=1 —-worker-class eventlet wsgi:app &
  ```

- (si nécessaire : arrêter le serveur gunicorn) :

  ```
  $ pkill gunicorn
  ```

L'équipe HydroViz

Julien Paris

Architecte DPLG, chercheur en sciences de l'information et de la communication (ex-doctorant CNRS en Turquie), M2 pro gestion de projets culturels (Paris I Panthéon La Sorbonne). Actuellement chercheur et développeur indépendant en data visualisation et data analysis

Site personnel : http://www.jpylab.fr/

Compétences informatiques :

Python (dev. web *full stack*, Pandas, géométrie analytique...)
Systèmes d'information géographiques (QGIS, uDig, …)
Design graphique (InkScape, suite Adobe, ...)
Design 3D (Blender, AutoCAD, ArchiCAD, …)
Javascript (D3, Leaflet), Méthodes agiles et développement continu (git)
Bases de données NoSQL (MongodB)

Centres d'intérêt :

data visualisation, open data, théorie des graphes,
résilience et métabolisme urbain, projet "fab city"
fablabs & fabrication collaborative, piles microbiennes DIY
mathématique des origamis, découpe CNC et laser

Langues étrangères : anglais (courant), espagnol (courant), turc (courant)

Contact : +33 6 83 65 84 91 | jparis.py@gmail.com | @jparis_py *(twitter)*

Missions sur le projet HydroViz :

data mining | conception | développement | déploiement
design graphique | UI-UX | formalisation

Florian Melki

Diplômé de l'UTC en IHM avec une spécialisation Philosophie, Technologie et Cognition. Actuellement ingénieur de recherche pour l'école Polytechnique de Nantes, spécialisé en data mining et data visualisation dans le cadre d'un projet ANR

Site personnel : https://www.linkedin.com/in/florian-melki-26842718

Compétences informatiques :

Dev. web *full stack* Javascript (nodeJs, D3, Leaflet...)
Base de données relationnelles (Sql Serveur, mySql) et NoSql (MongoDb)
Développement continu (git)
Conception 3D avec WebGl et Unity3D (débutant)
Fouille de données avec Tableau et Gephi
Gestion de projet

Centres d'intérêt :

data visualisation, musique (guitare, piano, MAO), DIY (montage de pédales pour guitare, projet personnel Arduino), sensibilité à l'upcycling et la lutte contre l'obsolescence programmée

Langues étrangères : anglais (courant), allemand (intermédiaire)

Contact : +33 6 33 88 00 64 | florian.melki@gmail.com | @FloDataviz *(twitter)*

Missions sur le projet HydroViz :

data mining | formalisation

SOMMAIRE

Démarche du projet HydroViz . 1

Principes de la data visualisation dans HydroViz . 2

Généralités 2

Le slider temporel 2

La carte interactive 3

Le treemap 5

Choix d'implémentation . 6

Conserver une ligne éditoriale 6

Contraintes de temps 6

Contraintes liées aux questions de propriété intellectuelle 6

Un projet en développement continu 6

Traitement des données et algorithmes . 7

Data mining / exploration préalable des données: 7

Prétraitement et de formatage préalable des données : 9

Données statistiques 9

Données géographiques (fonds de carte) 9

Intégration de bases de données supplémentaires 9

Dangerosité des produits 9

Eaux de surface 10

Sélection et analyse des données 10

Requêtes côté client 10

Analyse côté serveur 10

Représentation et échelles 10

Evolutions possibles . 12

Licences des logiciels utilisés . 13

Architecture de l'application 13

Design de l'application 13

Bases de données (sources) 14

Code source et hébergement 14

Documentation technique * . 15

Pré-requis techniques 15

Préparation des données / intégration de nouveaux jeux de données 15

Utilisation locale 15

Serveur physique (cloud) 16

L'équipe HydroViz . 18

Julien Paris . 18

Florian Melki . 18

SOMMAIRE . 19