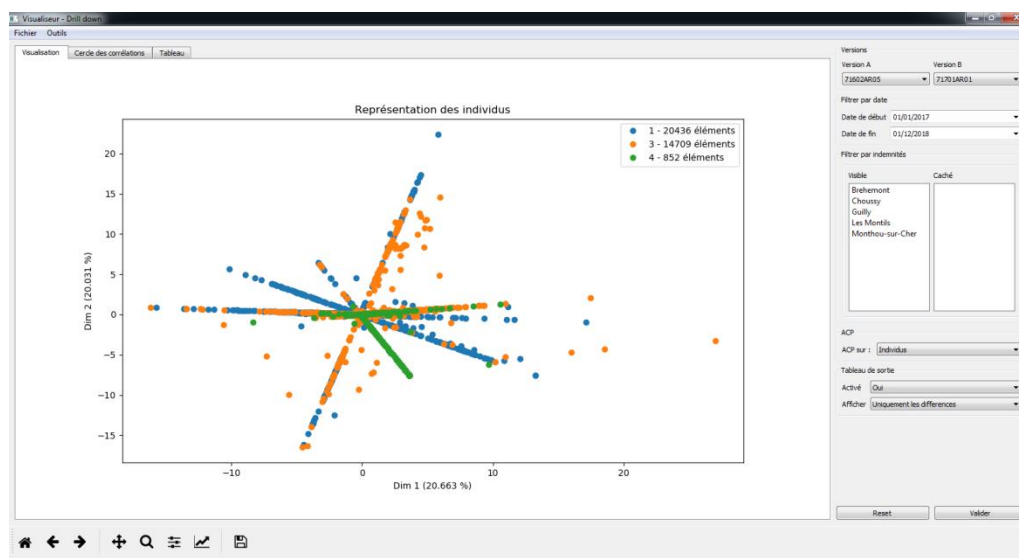


# Détection préventive de bugs dans le versement de la solde de l'armée

Projet de recherche et de développement  
2018 - 2019

**CONFIDENTIEL**



**Étudiant**  
Guillaume SERVAIS

**Tuteur académique**  
Nicolas RAGOT  
Gilles VENTURINI

**Tuteur entreprise**  
Mikaël WINANDY

## Liste des intervenants

### Entreprise

Sopra Steria  
47 Rue Christian Huygens  
37100 Tours  
www.soprasteria.com

Nom	Email	Qualité
Mikaël WINANDY	mikael.winandy@soprasteria.com	Tuteur entreprise

### École

École polytechnique de l'université de Tours  
64 Avenue Jean Portalis  
37200 Tours  
polytech.univ-tours.fr

Nom	Email	Qualité
Guillaume SERVAIS	guillaume.servais@etu.unit-tours.fr	Étudiant
Nicolas RAGOT	nicolas.ragot@univ-tours.fr	Tuteur académique
Gilles VENTURINI	gilles.venturini@univ-tours.fr	Tuteur académique

---

## Avertissement

Ce document a été rédigé par Guillaume SERVAIS susnommé l'auteur.

L'entreprise Sopra Steria est représentée par Mikaël WINANDY susnommé le tuteur entreprise.

L'Ecole Polytechnique de l'Université de Tours est représentée par Nicolas RAGOT et Gilles VENTURINI susnommés les tuteurs académique.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

L'auteur reconnaît assumer l'entière responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non-respect des lois ou des droits d'auteur.

L'auteur atteste que les propos du document sont sincères et assument l'entière responsabilité de la véracité des propos.

L'auteur atteste ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.  
L'auteur atteste que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

L'école polytechnique de l'université de Tours, l'entreprise Sopra Steria, ainsi que l'auteur, ne peuvent pas diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source sans l'autorisation écrite de chacune des trois parties.

## Pour citer ce document

Guillaume SERVAIS, Détection préventive de bugs dans le versement de la solde de l'armée, Projet Recherche & Développement, Ecole Polytechnique de l'Université François Rabelais de Tours, Tours, France, 2018-2019.

```
@mastersthesis{
  author={SERVAIS, Guillaume},
  title={ Détection préventive de bugs dans le versement de la solde de l'armée },
  type={Projet Recherche & Développement},
  school={Ecole Polytechnique de l'Université de Tours},
  address={Tours, France},
  year={2018-2019}
}
```

## Sommaire

Liste des intervenants .....	2
Avertissement .....	3
Pour citer ce document .....	4
Sommaire .....	5
Introduction.....	8
1. Contexte de la réalisation.....	9
1.1. Contexte .....	9
1.2. Enjeux .....	9
1.3. Objectifs.....	10
2. Description générale .....	11
2.1. Description de l'existant.....	11
2.1.1. Description générale .....	11
2.1.2. Les limites de l'existant .....	12
2.2. Environnement du projet .....	12
2.3. Caractéristiques des utilisateurs .....	12
2.4. Fonctionnalités du système.....	13
2.5. Structure générale du système .....	13
3. Etat de l'art .....	14
3.1. Méthodes de visualisation de données.....	14
3.1.1. Nuage de points.....	14
3.1.1. Courbe temporelle .....	16
3.1.2. Carte de densité .....	16
3.1.2.1. Carte thermique .....	17
3.1.2.1. Simplification d'un nuage de points .....	17
3.1.3. Graphe .....	18
3.1.4. Basé sur les pixels .....	18
3.1.5. Les arbres.....	19
3.1.5.1. L'arbre hiérarchique .....	19
3.1.5.2. L'arbre hyperbolique .....	20
3.2. Méthodes d'analyse des données.....	20
3.2.1. K-Means.....	21
3.2.2. Clustering hiérarchique .....	23
3.2.3. DBSCAN.....	23
3.3. Problèmes dans la détection d'anomalies .....	25

3.3.1.	Les types d'anomalies.....	25
3.3.1.1.	Anomalies ponctuelles .....	25
3.3.1.2.	Anomalies contextuelles .....	26
3.3.1.3.	Anomalies collectives .....	27
3.3.2.	Les problèmes à résoudre .....	27
3.3.2.1.	Le choix du seuil de décision .....	27
3.3.2.2.	L'identification d'une anomalie.....	28
3.3.2.3.	L'évolution de la définition de l'anomalie .....	28
3.3.2.4.	Le bruit dans les données .....	28
3.3.2.5.	Difficultés de généralisation.....	29
4.	Analyse et conception .....	30
4.1.	Visualisation .....	30
4.2.	Analyse des données .....	31
4.3.	Application.....	31
5.	Mise en œuvre.....	32
5.1.	Les objectifs de l'application .....	32
5.2.	Présentation de l'application .....	32
5.2.1.	Séquence d'utilisation .....	32
5.2.2.	Fenêtre principale : visualisation des indemnités .....	33
5.2.2.1.	Partie visualisation .....	34
5.2.2.2.	Partie filtrage .....	36
5.2.3.	Fenêtre secondaire : exploration des individus .....	37
5.3.	Structure de l'application .....	39
5.3.1.	Pattern MVC .....	39
5.3.2.	Diagramme UML.....	40
5.4.	Spécifications machine .....	40
5.5.	Les difficultés de la réalisation .....	41
6.	Démarche Qualité .....	42
6.1.	Les bibliothèques.....	42
6.1.1.	Matplotlib.....	42
6.1.2.	Scikit-learn .....	42
6.1.3.	Qt.....	42
6.2.	Tests.....	43
6.3.	GitLab .....	44
6.3.1.	Gestion des versions.....	44
6.3.2.	Tickets.....	44

---

6.3.3. Intégration continue.....	45
6.4. SonarQube.....	45
Bilan et conclusion.....	47
Table des illustrations.....	48
Bibliographie.....	49
Annexes .....	51
1. Description des interfaces externes du logiciel.....	53
2. Spécifications fonctionnelles.....	54
3. Spécifications non fonctionnelles.....	56
4. Comptes rendus hebdomadaires (Weekly).....	58
5. Document utilisateur.....	68
6. Document développeur.....	77
7. Rapport SonarQube.....	84
8. Diagramme de Gantt prévisionnel .....	85
9. Diagramme de Gantt réel.....	86

---

## Introduction

Ce document présente le projet de recherche et de développement en partenariat avec la société Sopra Steria. Ce projet, nommé « détection préventive de bugs dans le versement de la solde de l'armée » a pour but de faciliter la détection d'anomalies dans les résultats du logiciel de calcul des soldes lors de la mise à jour de celui-ci. En effet, chaque version du logiciel produit une grande quantité de données qu'il est impossible d'analyser à la main.

Le but de ce projet est donc d'automatiser une partie de l'analyse afin de simplifier la lecture des résultats pour l'opérateur en charge de contrôler la version du logiciel.

Le projet a été proposé à Polytech Tours par la société Sopra Steria qui est représentée par M. Mikaël WINANDY. L'étudiant en charge de la réalisation de ce projet est M. Guillaume SERVAIS et il est encadré par M. Nicolas RAGOT et par M. Gilles VENTURINI, tous deux enseignants chercheurs à Polytech Tours.

Ce rapport de fin de projet se décompose en plusieurs parties. Dans un premier temps, le rapport est consacré à expliquer le contexte du projet, puis à le décrire. Une partie consacrée à l'état de l'art est ensuite exposée, suivie d'une partie d'analyse et de conception. S'en suivra alors une partie détaillant la mise en œuvre ainsi que la démarche qualité qui a été mise en place.



# 1. Contexte de la réalisation

## 1.1. Contexte

Sopra Steria développe le logiciel destiné à calculer la solde des militaires. Ce logiciel se nomme LOUVOIS pour « Logiciel unique à vocation interarmées de la solde ». Lancé en 1996, il est entré en action en 2011. Il gère maintenant la solde d'environ 200 000 militaires en prenant en compte près de 300 indemnités différentes.

Depuis 2011, le logiciel évolue régulièrement pour corriger les bugs, implémenter les nouvelles règles de réglementation, etc...

Les versions sortent au rythme de :

- Une version majeure tous les 6 mois
- Une version intermédiaire tous les mois
- 4 à 5 versions dites "release" par mois

Dans ce contexte, chaque version doit être vérifiée/testée dans la semaine afin de détecter d'éventuelles anomalies. Ces vérifications doivent être très rapides de manière à ne pas prendre du retard sur les vérifications des versions suivantes.

Le temps d'exécution du calcul des soldes est d'environ deux jours. Donc pour pouvoir comparer les résultats avec une autre version (par exemple la version de production), il faut 2 \* 2 jours pour calculer les résultats des deux versions. Ce qui laisse 3 jours pour effectuer des analyses sur les résultats.

La maintenance du système LOUVOIS s'effectue dans la caserne Rannes de Tours. Elle mobilise près de 60 personnes qui sont segmentés en plusieurs groupes, chaque groupe ayant sa spécialité. Par exemple, il y a un groupe dédié à la gestion de la base de données, un groupe dans le développement de nouvelles fonctionnalités, un groupe pour tester et analyser les nouvelles versions, ...

## 1.2. Enjeux

Le test des nouvelles versions est un enjeu majeur. En effet, la moindre erreur peut engendrer des dizaines de milliers, voire des millions d'euros de non payé ou de trop payé. En plus du côté financier important, l'aspect social n'en est pas moins car une solde non versé correctement peut mettre en difficulté une famille entière.

Il a été estimé par la cour des comptes, qu'en 2012 il y a eu 465 millions d'euros d'erreurs de calculs (moins-perçus ou trop-perçus).

La fiabilité de l'application d'analyse d'anomalies est donc un enjeu très important afin de limiter au maximum les erreurs de calculs dans chaque nouvelle version du système LOUVOIS.

### 1.3. Objectifs

L'objectif de ce projet est de proposer une solution de détection d'anomalies (bugs) dans le calcul des soldes des militaires. Cette solution doit apporter un gain par rapport à la solution actuellement utilisée (détaillée dans la partie 2), tant au niveau de la quantité et de la qualité des détails que dans la vitesse d'analyse des résultats. L'opérateur devra pouvoir naviguer de manière fluide dans les résultats ce qui implique qu'une mise à jour de l'affichage des résultats ne doit pas prendre plus de quelques secondes.

Pour réaliser ces objectifs, on a trois axes d'exploration. Chaque axe se repose sur le précédent ce qui implique qu'ils doivent être réalisés, au moins en partie, dans l'ordre. Ces trois axes sont :

- Améliorer la procédure existante en travaillant sur l'ensemble des individus plutôt que sur leur agrégation, adapter les représentations graphiques, ...
- Proposer de nouvelles représentations graphiques avec une possibilité de filtrage pour masquer certaines indemnités, par exemple.
- Augmenter la quantité de données traitées en ajoutant des informations descriptives des individus pour faire apparaître les facteurs communs des individus d'un même groupe. Par exemple, le nombre d'enfant, le statut marital, ...
- Retrouver les facteurs à l'origine des différences entre les deux versions du logiciel. Par exemple, pour telle indemnité, il y a eu un fort changement pour les individus de plus de 30 ans et mariés.
- Prédire les individus et les facteurs qui seront susceptibles de poser des problèmes ou de révéler des anomalies lors de versions futures.

## 2. Description générale

Cette partie permet de décrire le fonctionnement du projet de façon générale en exposant ses caractéristiques, ainsi que son environnement d'utilisation. Dans cette partie, il est aussi décrit le fonctionnement de la solution existante.

### 2.1. Description de l'existant

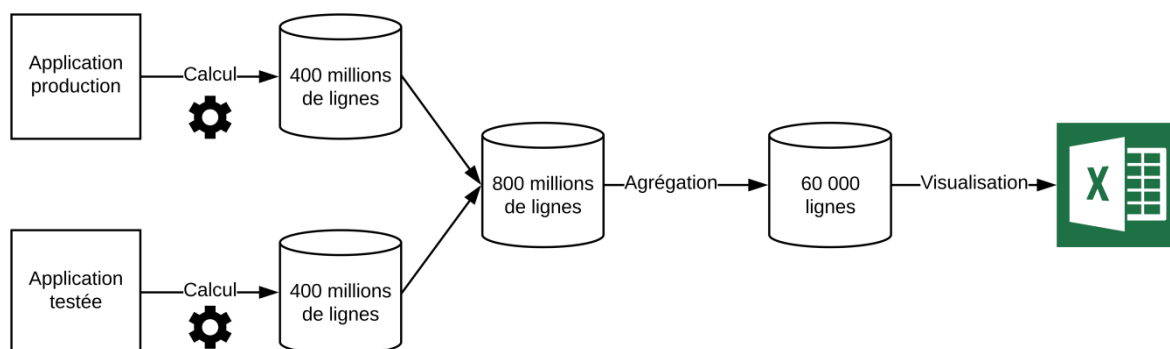
#### 2.1.1. Description générale

Afin de tester chaque nouvelle version, une comparaison entre la nouvelle version et la version de production est effectuée. Pour cela, deux calculs sont effectués sur le même jeu de données. Le premier permet d'obtenir les résultats avec la version de production et le second permet d'avoir les résultats calculés par la nouvelle version. Pour visualiser correctement les changements, les calculs se font sur 24 à 36 mois.

Les données obtenues sont les discriminants de régularisation<sup>1</sup> des soldes des militaires. Si le discriminant est égal à 0 c'est que la solde a été correctement réglée, sinon, cela signifie qu'il y'a un changement dans le calcul de l'indemnité. Ce changement peut être volontaire (une indemnité qui change de coefficient), mais il peut aussi provenir d'un bug.

Les résultats des calculs sont stockés dans une base de données Oracle. Chacune des deux versions de l'application génère environ 400 millions de lignes. Les 800 millions de lignes sont ensuite agrégées<sup>2</sup> en environ 60 000 lignes pour pouvoir faire des visualisations avec Excel. L'agrégation est obligatoire car Excel n'accepte pas une telle quantité de données. De plus, analyser les résultats des deux versions en comparant les lignes deux à deux est difficile à réaliser au vu du grand nombre de lignes générées en sortie.

Le schéma ci-dessous représente la chaine actuelle.



**Illustration 1 : Schéma du fonctionnement de la méthode d'analyse existante**

<sup>1</sup> Discriminants de régularisation : Différence entre ce qui a été payé par la version de production du logiciel et ce qui aurait dû être payé selon la nouvelle version.

<sup>2</sup> Agrégation : L'agrégation est une opération permettant de diminuer le nombre d'éléments en les regroupant par similarité. Selon le résultat désiré, l'agrégation peut se faire de différentes façons, comme par exemple la moyenne ou la somme des éléments agrégés.

Actuellement, avec Excel, il est notamment possible de consulter le temps de calcul, le nombre d'échec de calcul, les montants totaux à verser en plus ou en moins. Il est aussi possible de voir l'évolution, sur les 24-36 mois, des versements de chaque indemnité afin de voir s'il y'a des changements brusque ou s'il y a bien les comportements attendus.

Cette analyse doit cependant se faire à la main par un opérateur. Aussi, il n'est pas possible de porter son analyse sur des critères en particulier (ex. Sexe, nombre d'enfant, ...). Il n'est pas non plus possible d'explorer facilement les données lorsqu'une éventuelle anomalie est détectée. Ce travail doit se faire manuellement par l'opérateur en fouillant dans les données.

### 2.1.2. Les limites de l'existant

Avec Excel, l'analyse porte sur les 60 000 lignes issues de l'agrégation des 800 millions de lignes. La quantité de données étudiées est donc très inférieure à la quantité initiale. Cette grande différence provoque une forte perte d'informations. La forte agrégation peut donc masquer assez facilement certaines informations révélant de possibles dysfonctionnements de l'application (bugs).

De plus, l'analyse doit être faite manuellement ce qui fait que l'opérateur ne peut analyser, par manque de temps, que les quelques premiers plus gros écarts par rapport à la version de production. Lorsqu'une possible anomalie est détectée, sa classification (anomalie ou comportement normal) doit se faire à la main parmi les données de l'agrégation ce qui rend la tâche fastidieuse.

Aussi, il n'y a pas de seuil précis pour catégoriser une anomalie ce qui rend le jugement arbitraire. De plus, cette opération est effectuée par un humain qui n'est pas infallible et qui est contraint par le temps. Cela augmente donc la probabilité de ne pas remarquer une anomalie.

## 2.2. Environnement du projet

Ce projet nécessite l'accès aux résultats des calculs des versions testés du logiciel LOUVOIS. Ces résultats étant stockés dans une base de données Oracle, l'accès à cette base de données est donc indispensable. Cette base de données est hébergée sur un serveur distant de l'ordinateur exécutant l'application.

Face à la quantité de données analysée et pour des raisons de performances, il sera probablement nécessaire de faire une copie locale d'une partie des données sur la machine exécutant l'analyse. Cette machine devra donc être en capacité d'accueillir ces données et d'être suffisamment performante pour les analyser rapidement. Ses caractéristiques restent à définir selon les méthodes d'analyses qui seront choisies.

La machine accueillant l'application devra posséder le système d'exploitation Windows ou Linux en version 64 bits et devra avoir les spécificités physique décrites dans la partie 3.2.2 en annexe.

## 2.3. Caractéristiques des utilisateurs

Les utilisateurs qui utiliseront l'application possèdent de bonnes connaissances en informatique. Les utilisateurs visés sont ceux qui effectuent déjà l'analyse manuellement (via Excel). Ils ont donc aussi de très bonnes connaissances métier leur permettant d'interpréter les résultats.

De par le grand nombre de versions testées chaque mois, les utilisateurs seront des utilisateurs réguliers de cette application.

## 2.4. Fonctionnalités du système

Le système permettra de traiter puis de visualiser les données selon plusieurs représentations graphiques. L'utilisateur aura la possibilité de naviguer dans le graphique en se déplaçant ou en zoomant dedans. Il aura aussi la possibilité d'appliquer des filtres sur les indemnités et sur les individus pour masquer certaines données.

De même, l'utilisateur pourra sélectionner plusieurs points ou groupes de points afin de consulter les dénominateurs communs à ces points dans le but de comprendre les facteurs susceptibles d'influencer le calcul d'une indemnité.

## 2.5. Structure générale du système

Il est possible de schématiser le système avec l'illustration suivante :

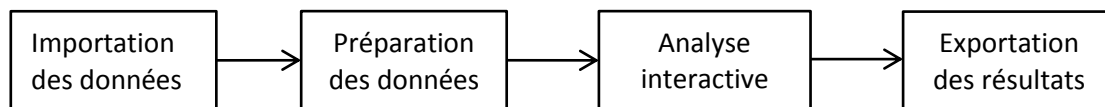


Illustration 2 : Structure générale du système

- Importation des données : Cette étape permet à l'application de récupérer les données à analyser depuis le système LOUVOIS.
- Préparation des données : Cette étape consiste à changer le formatage des données afin que celles-ci soient plus facilement manipulables par l'application. Cette opération a aussi pour but d'augmenter la pertinence des résultats affichés ainsi que de diminuer le temps de réponse de l'application.
- Analyse interactive : A cette étape, l'utilisateur interagit avec l'application pour explorer les données. Son exploration peut se faire en changeant la représentation graphique, en zoomant dans le graphique, en filtrant les données, ...
- Exportation des résultats : Cette étape permet à l'utilisateur d'exporter un rapport synthétisant les résultats.

### 3. Etat de l'art

Cette partie décrit différentes méthodes pouvant être utilisées pour répondre à la problématique du projet. Comme le projet requiert des méthodes de visualisation de données ainsi que des méthodes d'analyses pour pouvoir extraire des caractéristiques des individus, cette partie est divisée en deux pour détailler chacun des deux aspects. Une troisième partie est consacrée aux problèmes liés à l'étude d'anomalies.

#### 3.1. Méthodes de visualisation de données

La visualisation des données permet de réaliser un processus d'analyse de celles-ci. L'analyse graphique est différente d'une analyse effectuée par un algorithme car l'homme est capable de reconnaître des formes, déterminer des tendances, ... de façon très rapide et naturelle alors qu'effectuer les mêmes constatations via un algorithme est très compliqué, et parfois impossible. L'homme est donc utilisé en complémentarité des algorithmes.

Cependant, il faut faire attention à l'interprétation graphique que l'on peut faire des données. En effet, lorsqu'on a beaucoup de dimensions<sup>3</sup>, la représentation spéciale des données (initialement sur  $n$  dimensions) va être très déformée pour pouvoir obtenir une représentation graphique en 2 ou 3 dimensions. De ce fait, on peut apercevoir des formes qui, en réalité, n'existent pas dans les données. Il est aussi possible que tout un ensemble de données soit masqué par d'autres données. L'ordre du tracé du graphique influence donc beaucoup le graphique obtenu. La représentation graphique est donc à prendre avec critique et doit être complétée par une analyse algorithmique.

Il existe de multiples représentations graphiques. Je ne vais vous en présenter que quelques-unes, les plus importantes ainsi que celles qui peuvent amener à être utilisées dans le cadre du projet de recherche et de développement.

##### 3.1.1. Nuage de points

La représentation graphique avec un nuage de points est la plus connue car elle est l'une des représentations graphiques les plus faciles à mettre en place. Cette représentation graphique a aussi l'avantage de pouvoir se réaliser sur autant d'axes qu'il y a de dimensions dans les données.

Avec cette représentation graphique, on distingue assez facilement les 2 groupes

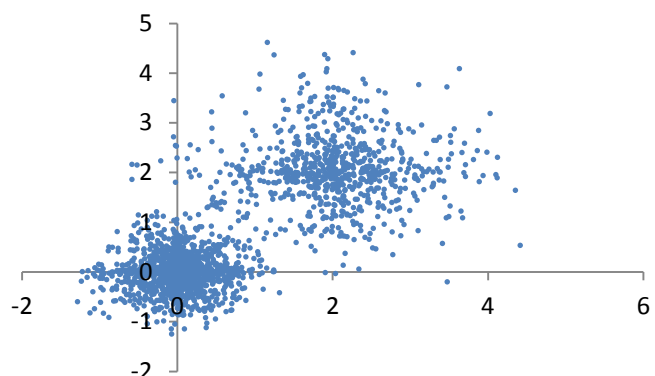


Illustration 3 : Représentation en nuage de points

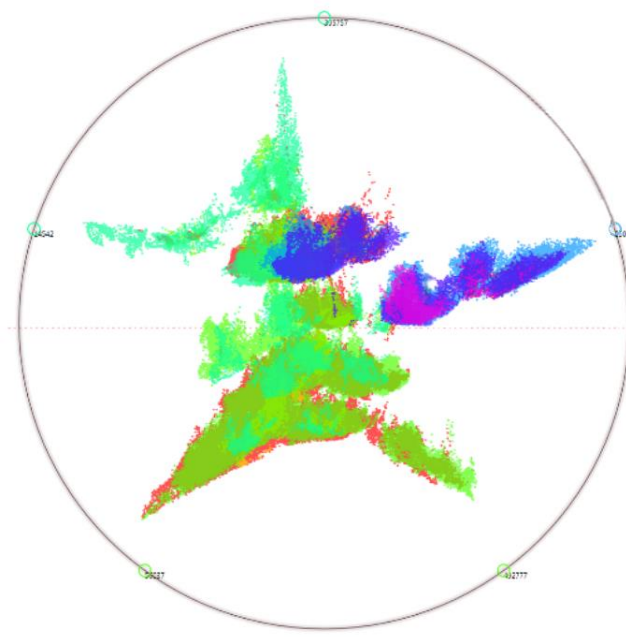
<sup>3</sup> Dimension : En analyse de données, une dimension correspond à une caractéristique permettant de définir un individu.

de points dans l'illustration 3. En revanche, on remarque que la densité de points est telle autour de l'origine du graphique (en 0, 0), qu'il y a un certain nombre de points qui sont masqués par les autres.

Pour représenter les nuages de points, on peut prendre un repère orthogonal comme dans l'illustration 3, mais on peut aussi faire une représentation radiale.

Pour la représentation orthogonale, on se sert des valeurs de chaque dimension pour positionner les individus. Lorsqu'il y a moins de dimensions représenté que d'attributs définissant les individus, il faut faire une réduction du nombre de dimensions. Pour effectuer cette tâche, il existe plusieurs algorithmes comme Multi-Dimensionnal Scaling (MDS), FastMap, ... Certaines sont plus rapides que d'autres, déforment plus ou moins les données, ... Par exemple, FastMap implique le même niveau de déformation des données que MDS mais effectue plus rapidement la réduction du nombre de dimension.

Pour la représentation radiale, on place les individus en fonction des distances des quelques individus choisis. L'avantage de cette représentation, est que selon les individus choisis, les nuages obtenus vont être très différents, et donc révéler des informations différentes. Cette technique a aussi l'avantage de pouvoir être calculée très rapidement sur un large lot de données.



**Illustration 4 : Représentation radiale**

### 3.1.1. Courbe temporelle

La représentation des données via une courbe est très utilisée notamment pour représenter des variations au cours du temps. Cette représentation permet de visualiser facilement les évolutions, chose qui n'est pas facile à faire lorsqu'on regarde un tableau de données brutes.

Par exemple, avec la visualisation ci-dessous, on remarque que le nombre d'objets connectés à internet (en milliard) augmente de manière exponentielle lorsqu'on prend un laps de temps de 10 ans. Aussi on remarque facilement qu'actuellement, il y a assez peu d'objets connectés par rapport à ce qui est estimé pour 2025.

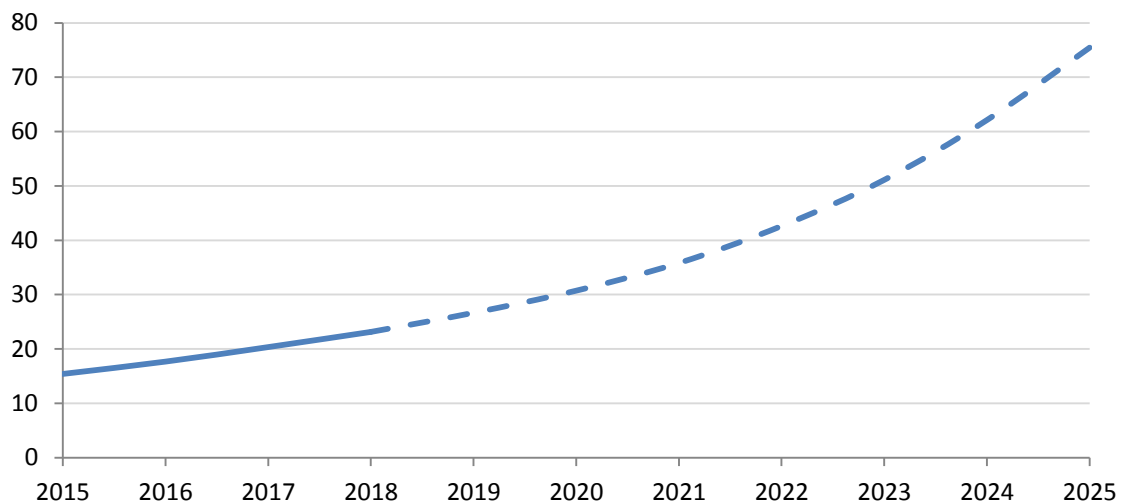


Illustration 5 : Evolution du nombre d'objets connectés

### 3.1.2. Carte de densité

Les représentations graphiques par densités sont utilisées pour représenter la répartition d'une densité dans un espace donné. Cette représentation est facilement identifiable car elle est composée d'un nuancier de couleurs.

Les cartes de densités sont utilisées pour représenter des fréquences d'évènements ou des densités d'évènement. Par exemple, avec ces cartes, on peut représenter les quantités de pluies, la densité du trafic routier d'une ville, ...



### 3.1.2.1. Carte thermique

Les cartes thermiques, ou « Heat map » en anglais, sont souvent utilisées pour représenter la répartition d'un évènement sur un espace géographique.

Par exemple, on peut représenter les pertes thermiques des habitations dans une ville ou encore les lieux de tournage de film à Paris comme dans l'illustration 6.

On remarque facilement, que le cœur de Paris, est le plus propice à accueillir un tournage de film. En revanche, plus on s'en éloigne, moins il y eu de films tournés.

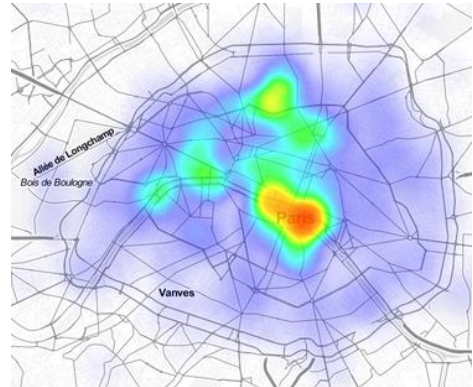


Illustration 6 : Carte thermique

### 3.1.2.1. Simplification d'un nuage de points

Les cartes de densités peuvent aussi être utilisées pour simplifier la représentation graphique des nuages de points lorsqu'il y a trop de points à représenter à un même endroit. Cela permet de ne pas se retrouver dans une situation où tout une zone du repère est remplie de la couleur des points et donc en deviens illisible.

Par exemple, sur l'illustration 7 sont représentées les localisations des bureaux de la société CGI. Sur la carte de gauche, on y aperçoit donc différents repère mais aussi deux bulles, une sur Paris et l'autre sur Lyon. En effet, avec le niveau de zoom actuel, les deux points sur Paris ou Lyon sont trop proche et il serait donc illisible de les afficher par conséquent, ils ont été regroupés. Cependant, lorsqu'on zoom sur Paris (carte de droite), par exemple, on obtient une résolution suffisante pour afficher ces points correctement et donc on les représente indépendamment.



Illustration 7 : Simplification d'un nuage de points

### 3.1.3. Graphe

La représentation graphique de données peut aussi être faite grâce à des graphes. Les graphes permettent de visualiser les relations entre les individus. Ils sont constitués de sommets (les individus) et d'arc (les relations entre les individus).

Les graphes sont très utilisés dans le domaine des réseaux sociaux pour représenter les relations entre les différents membres.

Les graphes peuvent être représentés en 2 ou 3 dimensions. Souvent, ils auront une forme quelconque (à gauche de l'illustration 8), mais ils peuvent aussi prendre des formes plus originales dans certaines conditions. Par exemple, on peut avoir une forme de ballon comme au centre de l'illustration 8, ou de tore, à droite.

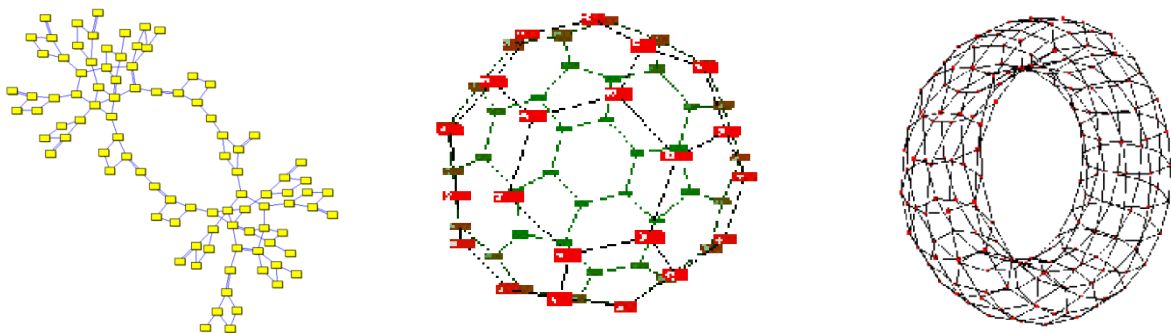


Illustration 8 : Graphe en 2 et 3 dimensions

### 3.1.4. Basé sur les pixels

Cette méthode a pour but d'attribuer une couleur pour chaque individu en fonction de ses attributs. Cela permet de constater visuellement la prédominance de certains attributs ainsi que leurs répartition dans l'espace.

Le principe de fonctionnement est plutôt simple car il repose sur la même technique d'attribution de la couleur que les pixels d'un écran LCD. Chaque individu est représenté par un pixel. Chaque pixel est divisé en sous pixels qui correspondent aux caractéristiques des individus. Chaque sous pixel a une couleur attribué et va participer plus ou moins en fonction de la valeur de l'attribut pour l'individu. Un individu est donc représenté comme le montre l'illustration 9.

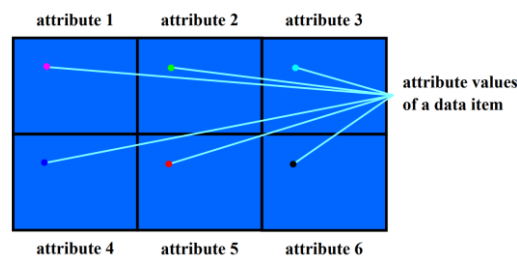


Illustration 9 : Représentation d'un individu

Ensuite pour tracer les pixels sur l'écran il y a plusieurs technique comme commencer à tracer dans le centre de l'écran puis faire une spirale en ayant rangé les pixels par couleurs, ou encore les placer dans un ordre quelconque. L'illustration 10 montre un exemple de rendu avec un tracé en spirale combiné avec une technique de segmentation de l'écran.

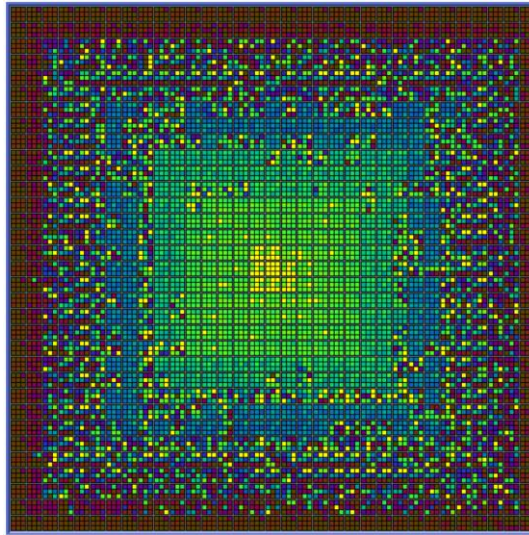


Illustration 10 : Exemple de rendu avec une représentation basé sur les pixels

### 3.1.5. Les arbres

Les arbres permettent de représenter et de mettre en évidence une hiérarchie entre les individus. On peut représenter ces arbres de différentes façons mais ils sont généralement représentés de façon hiérarchique comme pour un arbre généalogique, ou de façon hyperbolique.

#### 3.1.5.1. L'arbre hiérarchique

Les arbres hiérarchiques sont simples à tracer mais deviennent vite illisibles lorsqu'il y a beaucoup de niveau et d'individus par niveau. On utilise donc majoritairement ces arbres pour représenter de petites quantités de données.

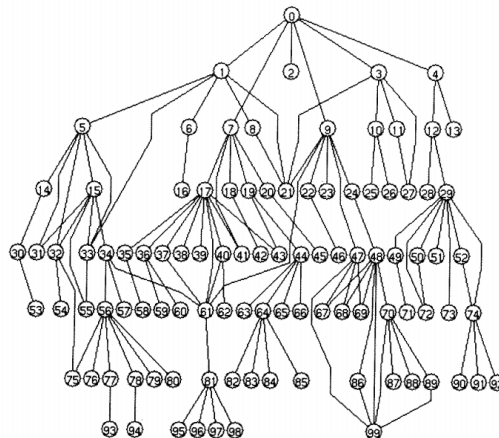


Illustration 11 : Arbre hiérarchique

### 3.1.5.2. L'arbre hyperbolique

Les arbres hyperboliques s'affranchissent davantage de la limitation du nombre d'individus. En effet, plus l'individu est loin de la racine de l'arbre, plus il est représenté en petit et est donc « masqué ». De plus avec cette représentation, on aperçoit assez facilement les branches les plus importantes, ce qui n'est pas le cas dans la représentation hiérarchique.

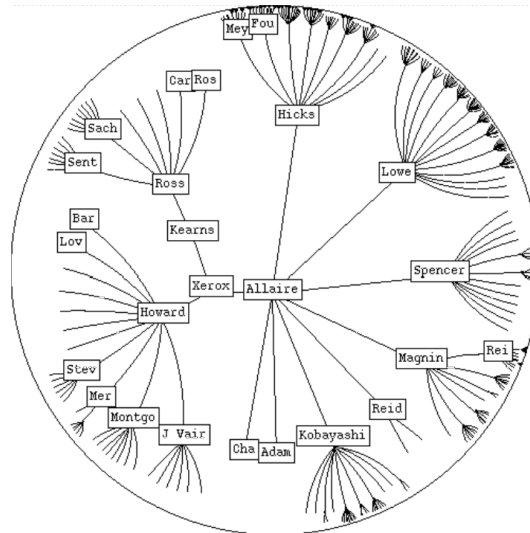


Illustration 12 : Arbre hyperbolique

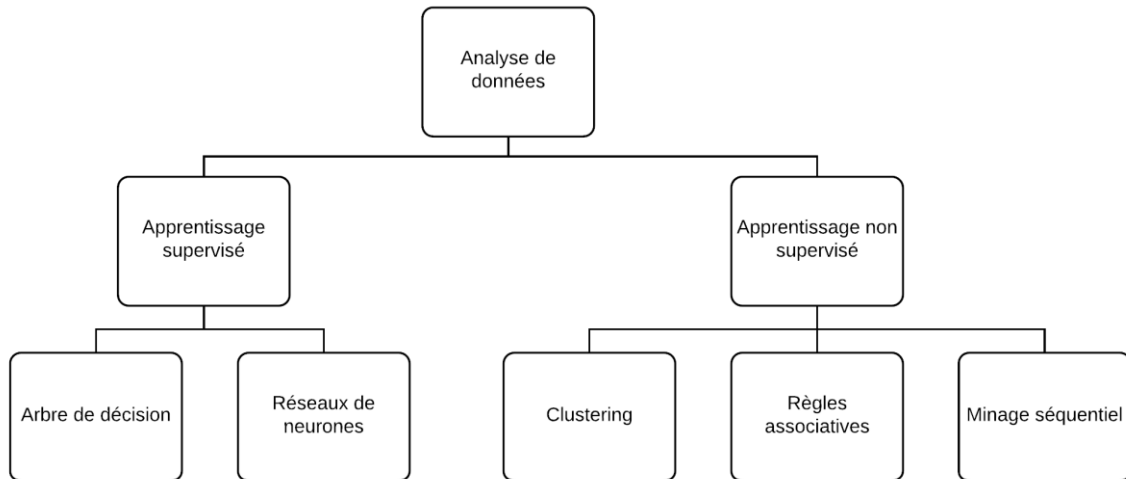
## 3.2. Méthodes d'analyse des données

L'analyse de données, « data mining » en anglais, a pour but de faire ressortir des informations sur une grande quantité de données. Elle est complémentaire à l'analyse graphique des données car elle permet d'obtenir des indicateurs statistiques, des tendances, et même des relations entre les données. L'analyse de données ne fournit pas de solution toute faite mais fournit un panel de résultats qu'il faut ensuite interpréter en fonction du problème étudié.

Il existe deux grandes catégories dans l'analyse de données : la classification supervisée et la classification non supervisée. La classification supervisée permet de classer un nouvel individu en fonction d'une base d'apprentissage. Il est aussi possible de prédire ou d'estimer des valeurs manquantes grâce au modèle construit précédemment. La classification non supervisée est utilisée pour identifier des ensembles d'éléments qui partagent des similarités. Son fonctionnement ne requiert pas de base d'apprentissage et ne permet donc pas de classer les éléments selon des classes prédéfinies. Cette classification ne peut donc pas faire de prédiction mais se contente d'explorer les données pour trouver les patterns intéressants.

Dans l'apprentissage supervisé, on retrouve les algorithmes d'arbre de décision (CART, OC1, SLIQ, ...) ainsi que les réseaux de neurones (AdaBoost, Learn++, ...).

Pour l'apprentissage non supervisé, on a des algorithmes de clustering<sup>4</sup> (C.hierachique, K-means, EM, ...), de règles associatives (Apriori, ECLAT, SSDM, ...), et de minage séquentiel (GSP, SPADE, ...).



**Illustration 13 : Schéma de classification des algorithmes d'analyse de données**

Pour ce projet de recherche et de développement, nous allons nous concentrer sur des algorithmes de clustering qui peuvent très bien répondre aux objectifs du projet.

### 3.2.1. K-Means

L'algorithme des K-moyennes (K-means en anglais), est un algorithme non supervisé qui permet de déterminer des classes dans des données. Les classes obtenues n'ont pas de relation hiérarchique, c'est-à-dire, qu'il n'y a pas de classe incluse dans une autre, elles sont toutes indépendantes les unes des autres.

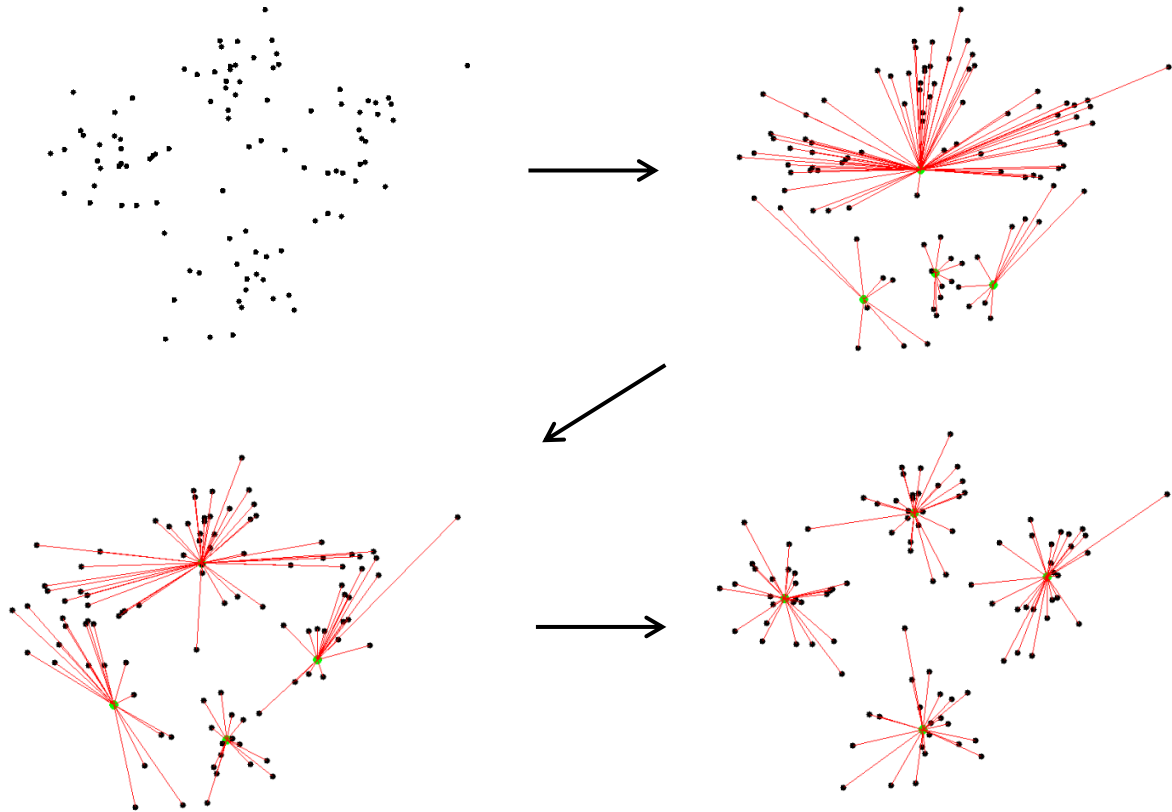
L'algorithme va segmenter les données en un nombre de classes définies avant son exécution. Lors de son lancement, l'algorithme va choisir k centre de classes. Il va ensuite attribuer une classe aux points les plus proches des centres des classes. L'algorithme va chercher à réduire la distance entre chaque point d'une même classe (distance intra-classe) tout en augmentant au maximum la distance entre chaque classe (distance inter-classe). Son algorithme est le suivant :

Choisir aléatoirement k centres de classes  
 REPETER  
 | Pour chaque classe, affecter le point le plus proche du centre de la classe a cette même classe  
 | Calculer le nouveau centre de chaque classe  
 TANT QUE les centres des classes bougent

<sup>4</sup> Clustering : regroupement de lots homogènes de données.



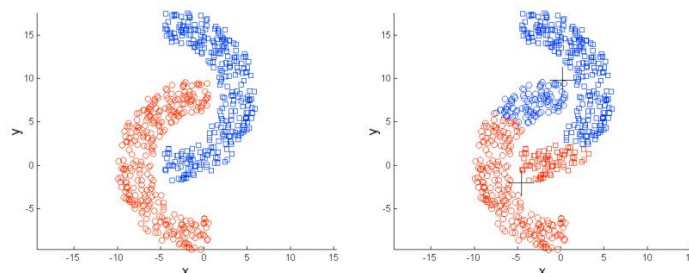
Dans l'illustration ci-dessous, on a les différentes itérations de l'algorithme avec en vert les centres des 4 classes. On remarque que les centres évoluent au court du temps. On remarque aussi que les distances intra-classes diminuent au fil des itérations.



**Illustration 14 : Evolutions du calcul du K-Means**

Les inconvénients de cet algorithme sont :

- Il faut fixer à l'avance le nombre classes souhaitées. Il est souvent difficile de connaître à l'avance le nombre de classes optimales, il faut donc faire plusieurs essais.
- Les points isolés sont mal gérés : à quelle classe doivent-ils appartenir ?
- Le résultat final n'est pas toujours le résultat optimal aux données car il est sensible à l'initialisation aléatoire des centres.
- Lorsque les données sont proches et quelles sont imbriquées, la classification peut être faussée comme dans l'exemple-ci-dessous.



**Illustration 15 : Limitation du K-Means**

### 3.2.2. Clustering hiérarchique

Le clustering hiérarchique a pour objectif de construire des classes imbriquées les unes dans les autres. L'avantage par rapport au K-Means c'est que l'on n'a pas besoin de préciser le nombre de classes voulues, on peut obtenir n'importe quel nombre de cluster en « coupant » l'arbre au niveau souhaité.

Il y a deux façons de construire l'arbre de hiérarchie : partir des individus et les regrouper jusqu'à obtenir un seul cluster (méthode agglomérative), ou faire l'inverse, partir d'un seul cluster et le diviser jusqu'à n'avoir que des clusters contenant qu'un seul individu (méthode par division).

Son algorithme pour la méthode agglomérative est le suivant :

Initialiser un cluster à chaque point  
REPETER  
| Fusionner les 2 clusters les plus proches  
TANT QUE le nombre de cluster est supérieur à 1

Et voici le rendu avec 6 individus :

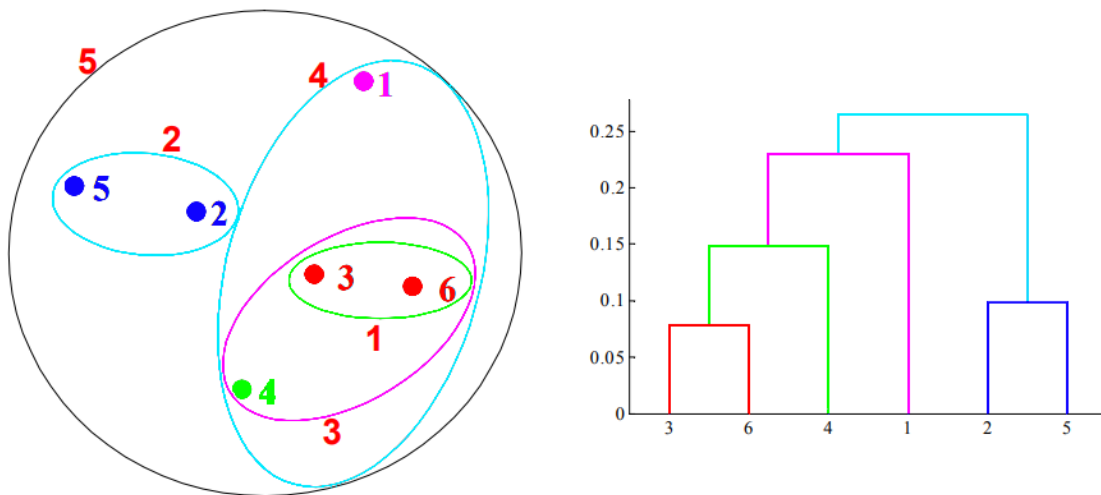


Illustration 16 : Clustering hiérarchique

### 3.2.3. DBSCAN

DBSCAN est un algorithme de clustering par densité. C'est-à-dire qu'il faut un certain nombre de points dans un rayon donné pour pouvoir considérer qu'un point fait partie d'une classe. Ce principe de fonctionnement permet de s'affranchir du bruit contenu dans les données. De plus, il n'est pas sensible aux formes comme peut l'être la méthode du K-Means. Cet algorithme a aussi l'avantage de pouvoir distinguer des classes lorsqu'elles sont superposées mais qu'elles ont une densité de points différente.

Avec DBSCAN, on peut obtenir les résultats suivant (à gauche sont représentés les points initiaux, à droite les mêmes points classés par DBSCAN) :

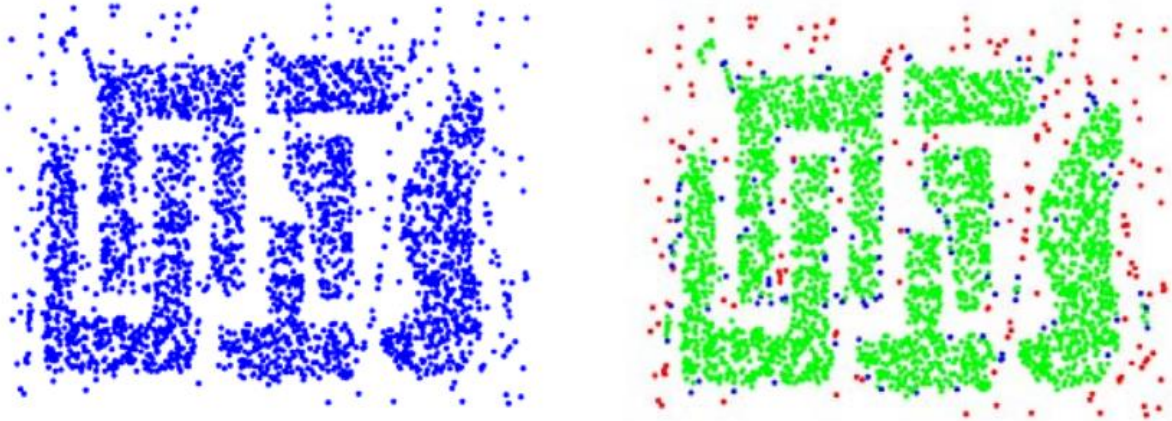


Illustration 17 : DBSCAN détection du bruit

Avec cette illustration, on remarque que DBSCAN arrive très bien à éliminer le bruit. En effet, les données utiles sont identifiées en vert alors que le bruit a été identifié en rouge et bleu.

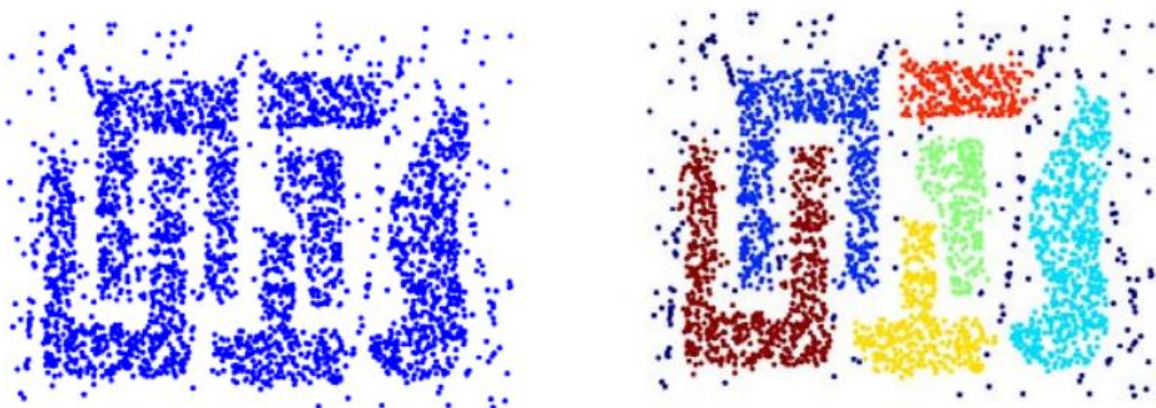


Illustration 18 : DBSCAN détection des classes

Avec cette illustration, on identifie parfaitement les différentes classes malgré leur forte imbrication. Aussi le bruit a été identifié et coloré en bleu très sombre.



Dans l'illustration ci-dessous, est comparé la clusterisation de points aléatoire par la méthode du K-Means et de DBSCAN. A gauche on a les points classés via la méthode du K-Means et a droite le classement par DBSCAN. On remarque que les classifications sont différentes et que la classification de DBSCAN est légèrement meilleure car DBSCAN a mis en évidence des points anormaux en rouge.

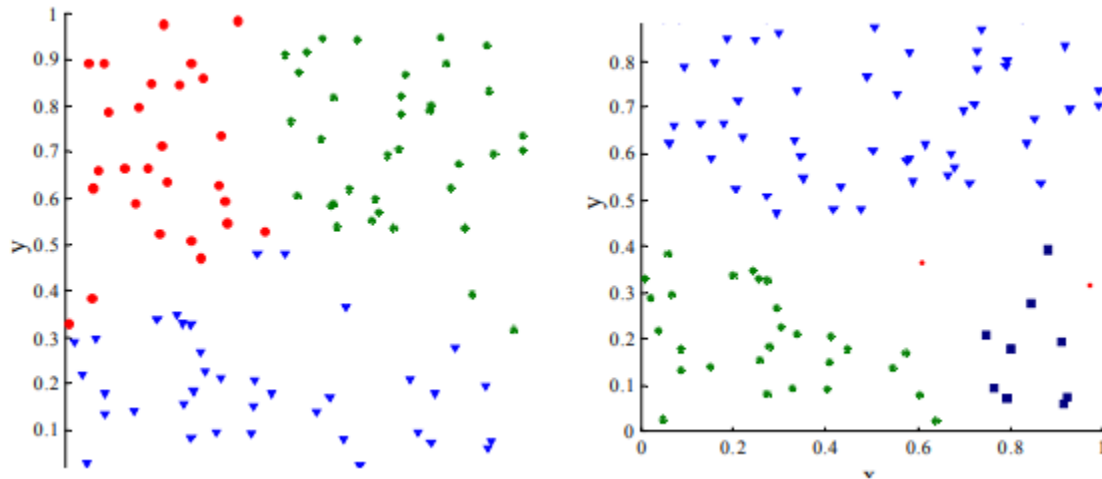


Illustration 19 : Comparaison K-Means - DBSCAN

### 3.3. Problèmes dans la détection d'anomalies

Avant de se lancer dans la détection d'anomalies, il faut d'abord définir ce qu'est une anomalie. Une anomalie est un écart par rapport à la normale, ou à la valeur théorique. Elle dépend donc de la situation. Par exemple, une variation de la température corporelle d'un individu peut être considérée comme anormale alors qu'en finance, une variation de la bourse est une situation normale.

#### 3.3.1. Les types d'anomalies

Il existe trois types d'anomalies. On peut être amené à en détecter plusieurs types simultanément mais cela augmente la complexité de l'algorithme de détection.

##### 3.3.1.1. Anomalies ponctuelles

Une donnée est considérée comme anomalie ponctuelle lorsque qu'elle n'est pas en adéquation avec le reste des données. Dans ce cas, on compare une donnée par rapport aux autres pour déterminer si elle est anormale ou non.

Par exemple, dans le graphique ci-après, on a un nuage de points (en bleu) réparti entre -1.5 et 1.5. Les points gravitant autour de ce nuage peuvent être considérés comme normaux. En revanche, les points en rouge sont très éloignés du nuage et sont donc considérés comme anormaux.

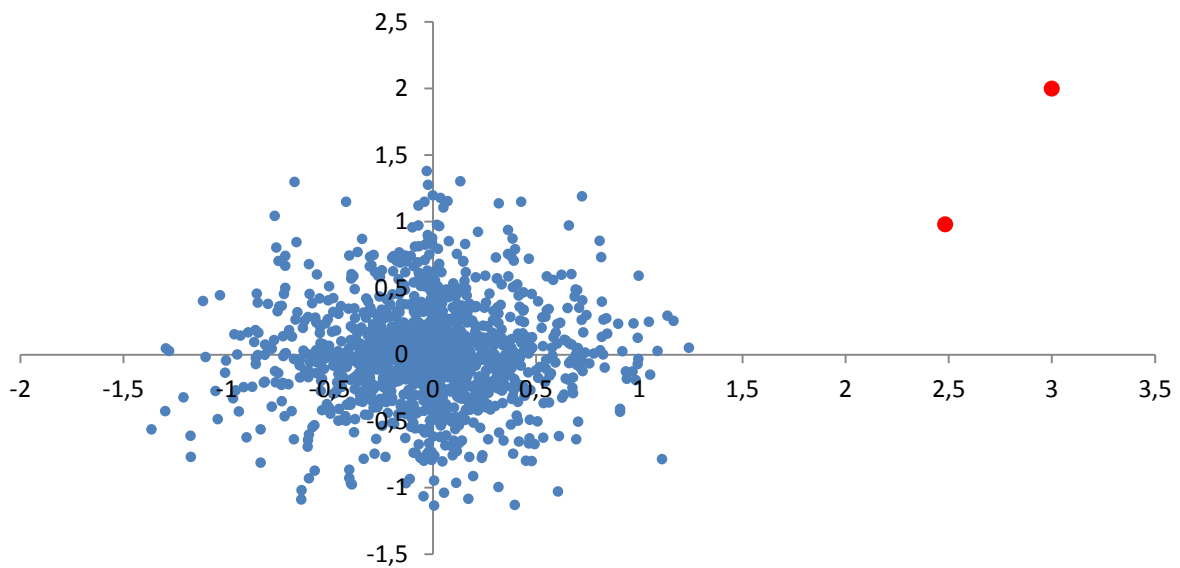


Illustration 20 : Représentation d'anomalies ponctuelles

### 3.3.1.2. Anomalies contextuelles

Dans le cas d'une anomalie contextuelle, une donnée peut être considérée comme normale ou anormale. Il faut donc connaître le contexte pour pouvoir la classer.

Par exemple, le graphique ci-dessous représente les relevés de températures extérieures à Paris. On remarque qu'en décembre il fait globalement plus froid qu'en aout. Cependant on remarque que le relevé en rouge est anormalement bas pour un mois de juin. Le relevé en rouge est donc une anomalie au vue du contexte.

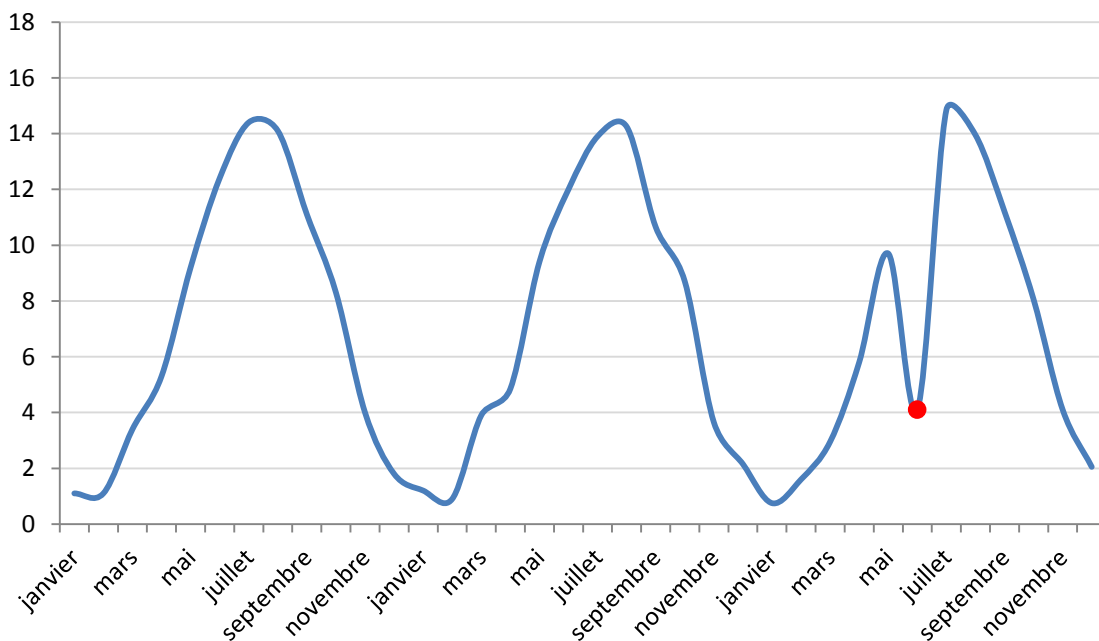


Illustration 21 : Représentation d'une anomalie contextuelle

### 3.3.1.3. Anomalies collectives

Une anomalie collective est identifiable par un groupe de données qui est anormal par rapport aux autres données. La particularité d'une anomalie collective est qu'elle concerne plusieurs points au lieu d'un seul comme c'est le cas dans les autres types d'anomalies.

On peut illustrer cette anomalie avec le graphique suivant. Le graphique peut représenter la tension aux bornes d'une prise électrique par exemple. La tension d'une prise évolue de façon sinusoïdale avec un rythme régulier. Sur la zone en rouge, on remarque que la courbe n'obéit plus à la forme sinusoïdale mais que tous les points sont à 0, ce qui peut signifier qu'il y a eu une microcoupure de courant ce qui n'est pas normal. Cet ensemble de points constituent donc une anomalie collective.

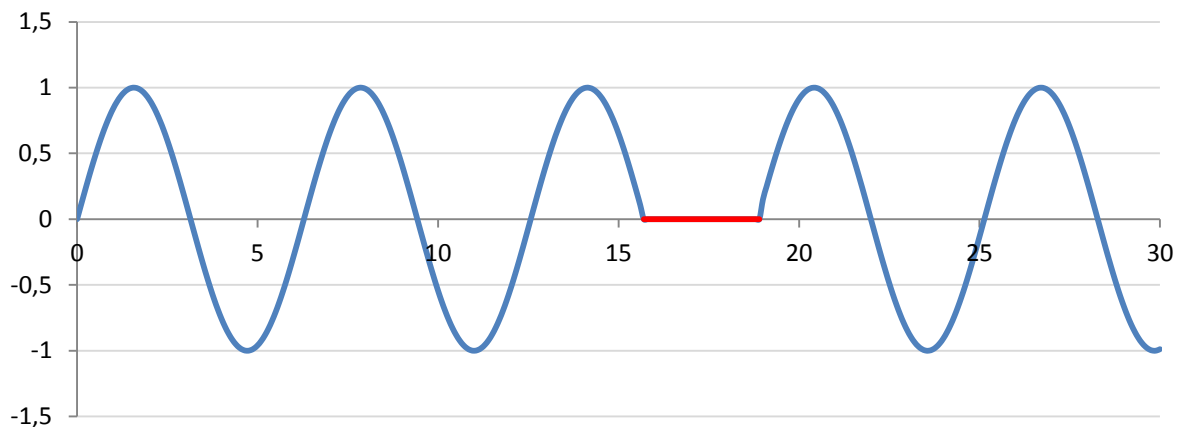


Illustration 22 : Représentation d'une anomalie collective

### 3.3.2. Les problèmes à résoudre

Pour faire de la détection d'anomalies, il faut principalement surpasser cinq problèmes. La résolution de ceux-ci peut être difficile et longue mais il est essentiel de les résoudre pour avoir une détection la plus fine et la plus juste possible.

#### 3.3.2.1. Le choix du seuil de décision

La définition d'un seuil de décision est essentielle pour pouvoir classer un élément de normal ou d'anormal. La difficulté réside dans le fait qu'il peut être difficile de choisir un bon seuil. Par exemple, si une pièce a ses dimensions qui varient de 5% par rapport aux dimensions attendus, est-ce un problème ? Et si elle varie de 10% ?

Une seconde problématique doit être traitée lorsque qu'un élément est très proche du seuil. Quelle est la décision que l'on doit prendre lorsque, par exemple, le seuil est à 5% et qu'on mesure 4.9% ou 5.1% ?

### 3.3.2.2. L'identification d'une anomalie

Il est parfois difficile d'identifier une anomalie présente dans le reste des données. Son identification peut être délicate lorsqu'elle est très similaire aux données normales.

Par exemple, lorsqu'un pirate souhaite attaquer un système informatique, il va essayer de se rapprocher le plus possible du fonctionnement normal du système afin de ne pas être détecté comme anomalie par des systèmes de protection qui pourrait le bloquer.

### 3.3.2.3. L'évolution de la définition de l'anomalie

La définition d'une anomalie peut évoluer avec le temps. En effet, les systèmes sont rarement figés et évoluent avec le temps ce qui implique qu'une anomalie peut devenir normale et qu'un élément normal peut devenir une anomalie.

Par exemple, dans les années 1900, il était anormal d'avoir une température supérieure à 10°C au mois de février en France. En 2050, avec le réchauffement climatique, ce seuil de température peut devenir normal et donc le nouveau seuil de température anomal pourra être de 15°C.

### 3.3.2.4. Le bruit dans les données

Les données analysées peuvent contenir du bruit et donc perturber les résultats. Le bruit est un signal parasite qui se rajoute au signal utile. Le bruit est aléatoire. La difficulté est d'en déterminer l'origine ainsi que sa plage de valeur, pour pouvoir différencier les éléments anormaux du bruit.

La représentation graphique suivante représente en rouge le signal d'origine et en bleu le signal avec du bruit. dans le signal bleu, on repère bien la présence du signal rouge.

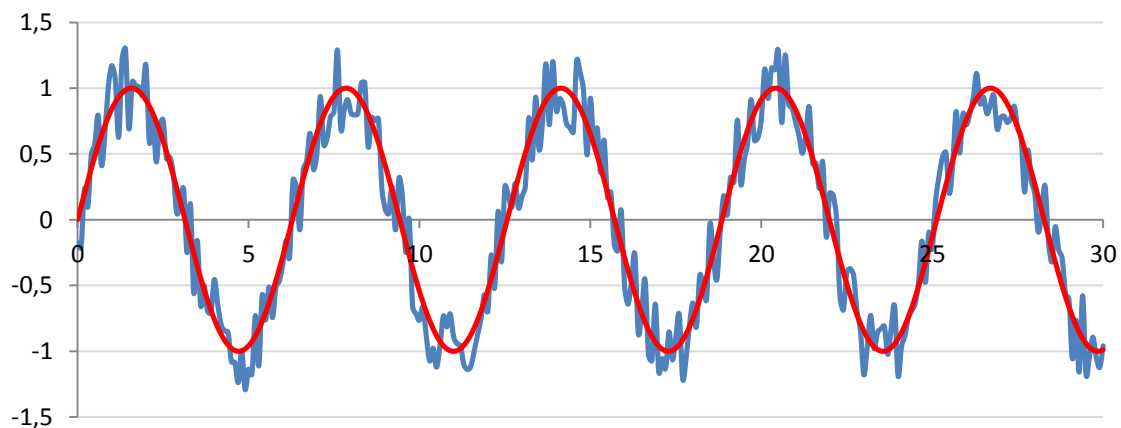


Illustration 23 : Représentation de données bruitées

---

### 3.3.2.5. Difficultés de généralisation

Il est difficile de généraliser un algorithme de détection d'anomalies car les anomalies elles-mêmes dépendent du problème considéré. En effet, dans certains cas, une situation normale correspond à des relevés qui ne varient que très peu (par exemple la température corporelle), alors que dans d'autres cas, les données varient normalement beaucoup (par exemple la bourse).

## 4. Analyse et conception

Dans cette partie, nous allons nous intéresser à appliquer au projet les méthodes étudiées lors de la partie de l'état de l'art.

Le principal défi du projet est de pouvoir déceler des anomalies qui ne sont pas identifiables avec la procédure appliquée actuellement. De plus, le but est de pouvoir fouiller dans toutes les données, et pas seulement dans des agrégations, pour réussir à trouver les quelques éléments anormaux noyés dans la grande quantité de données.

Face à ces quantités de données, il faut des méthodes d'analyse et de visualisation qui soient rapides à calculer, et les moins gourmandes possibles en quantité de mémoire vive. Nous allons étudier la possibilité d'appliquer les méthodes vues dans l'état de l'art dans les deux prochaines sous parties.

### 4.1. Visualisation

Pour visualiser la répartition des individus les uns par rapport aux autres, il est possible d'utiliser la représentation par nuage de points. Cependant, au vu de la quantité de données à afficher, il faudra combiner la représentation du nuage points avec la représentation par densité (comme la carte thermique). La visualisation obtenue fera donc apparaître la densité d'individus répartis dans l'espace.

Le principal problème de cette représentation est la complexité de calcul pour créer cette visualisation. En effet, les individus sont composés de plus de 300 attributs qu'il faut réduire en 2 ou 3 dimensions pour pouvoir les représenter graphiquement ce qui représente un certain volume de calculs.

La visualisation avec une courbe temporelle peut être utilisée pour obtenir l'évolution au cours du temps d'une indemnité. L'avantage de cette visualisation est qu'il est possible d'afficher plusieurs courbes sur le même repère pour pouvoir comparer les indemnités entre-elles.

La représentation par pixel peut aussi présenter un aspect intéressant, bien qu'il sera difficile d'interpréter les résultats sans avoir réduit préalablement le nombre de dimensions des individus.

Les représentations basées sur les arbres ou sur les graphes peuvent aussi être intéressantes pour montrer des relations entre les individus. Cependant, il est peu probable qu'elles soient utilisées dans ce projet car elles correspondent à des objectifs assez lointains.

On a vu dans cette partie que dans un premier temps, la représentation par densité sera privilégiée pour représenter les données. De plus, il faut prendre en compte que l'utilisateur devra pouvoir interagir avec la visualisation, donc il faut que celle-ci soit très facilement calculable pour qu'elle puisse être actualisée en quelques millisecondes.

Dans les actualisations possibles, l'utilisateur pourra zoomer/dé-zoomer, mais aussi filtrer les éléments représentés.

## 4.2. Analyse des données

En complémentarité avec la visualisation, l'analyse de données est importante. Par exemple, il est possible de faire des opérations de clustering sur les données, avant de les afficher. Cela permet par exemple, de coloriser la visualisation en faisant apparaître des groupes de points de la même couleur.

De plus, les données statistiques issues de l'analyse peuvent compléter les observations de visualisation et aider l'utilisateur à comprendre la répartition des individus.

Pour le projet, la méthode du K-Means ou du DBSCAN peuvent être intéressantes. En revanche, le clustering hiérarchique à moins d'intérêt.

Il est intéressant d'implémenter les deux premières méthodes pour laisser le choix à l'utilisateur d'utiliser la méthode qu'il préfère utiliser et selon ce qu'il souhaite explorer. Il est en effet très probable que les individus soient répartis dans l'espace avec des formes particulières, et dans ce cas seul DBSCAN arrivera à correctement faire les bons regroupements.

Pour répondre à l'objectif de prédire les individus qui peuvent potentiellement être problématiques pour une future version, l'utilisation d'un réseau de neurones est probablement indispensable. Cependant, cet objectif n'est pas prioritaire et demande encore beaucoup de recherches pour déterminer son application dans le projet.

## 4.3. Application

L'application met à disposition de la visualisation et de l'analyse de données. Elle est constituée d'une interface graphique avec laquelle l'utilisateur interagit. L'application est aussi au centre des calculs d'analyses des données.

L'utilisateur a la possibilité de filtrer les éléments qu'il souhaite analyser et/ou visualiser.

## 5. Mise en œuvre

Cette partie permet de décrire la partie réalisation du projet. Dans un premier temps, je vais vous présenter les objectifs de l'application développée puis dans un deuxième temps je vais vous présenter l'application ainsi que son utilisation, puis je vais vous présenter la structure du programme. Enfin je vais terminer par vous présenter les difficultés que j'ai rencontré lors du développement ainsi que les spécifications matérielles et logicielles que l'application nécessite.

### 5.1. Les objectifs de l'application

L'objectif de l'application est de reproduire et d'étendre la méthode de détection d'anomalies actuellement mise en place par la société Sopra Steria.

Pour réaliser cela, l'application doit être en mesure de proposer une visualisation des différences des valeurs des indemnités entre les deux versions du logiciel LOUVOIS testées. Cette visualisation doit pouvoir se faire via les valeurs numériques. Elle doit aussi pouvoir donner lieu à la sélection d'indemnités pour lesquelles, l'utilisateur souhaite obtenir plus d'informations.

L'utilisateur doit ensuite pouvoir visualiser l'évolution des valeurs dans le temps des indemnités sélectionnées. Cette visualisation doit avoir une granularité temporelle minimum à un mois.

Aussi afin de pouvoir comprendre et expliquer l'origine des différences entre les deux versions de LOUVOIS, il a été jugé nécessaire de pouvoir explorer les individus qui composent les indemnités sélectionnées. L'application doit donc être en mesure de réaliser cela.

### 5.2. Présentation de l'application

Dans cette section, je vais vous présenter l'application. Cette présentation a pour but de comprendre la procédure d'utilisation de l'application, mais aussi de connaître la signification des différents éléments graphique.

#### 5.2.1. Séquence d'utilisation

L'application a pour objectif d'être simple et rapide à utiliser. Pour cela, il a été décidé de manipuler des données synthétiques puis de pouvoir explorer les données avec une granularité de plus en plus fine.

Pour réaliser cela, l'application propose de visualiser l'évolution des différences des indemnités dans le temps. A partir de cette visualisation, l'utilisateur peut régler des filtres afin de restreindre sa visualisation à un nombre d'indemnités données ainsi que sur un laps de temps réduit. La zone sélectionnée est donc une zone qui présente des caractéristiques particulières sur lesquelles l'utilisateur souhaite en savoir davantage.



L'utilisateur va ensuite pouvoir visualiser la répartition des individus (militaires) dans l'espace en fonction de leur participation pour chaque indemnité sélectionnée. A partir de cette visualisation, il va pouvoir explorer les données et en déduire certains comportements.

Le schéma ci-dessous permet de résumer la séquence à suivre dans l'utilisation de l'application.

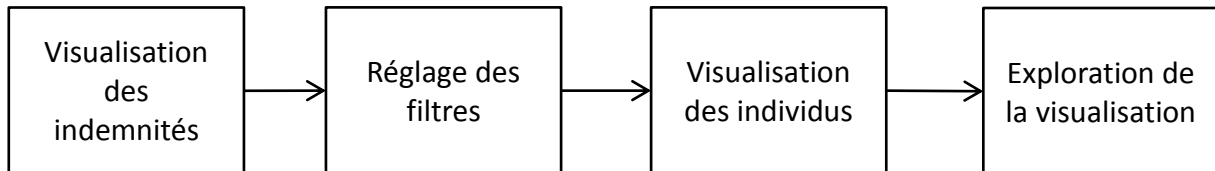


Illustration 24 : Séquence d'utilisation de l'application

### 5.2.2. Fenêtre principale : visualisation des indemnités

La fenêtre principale est la première fenêtre qui s'ouvre au lancement de l'application. C'est à partir de cette fenêtre que l'utilisateur va pouvoir visualiser l'évolution des indemnités dans le temps.

La capture d'écran ci-dessous montre l'interface de la fenêtre principale de l'application.

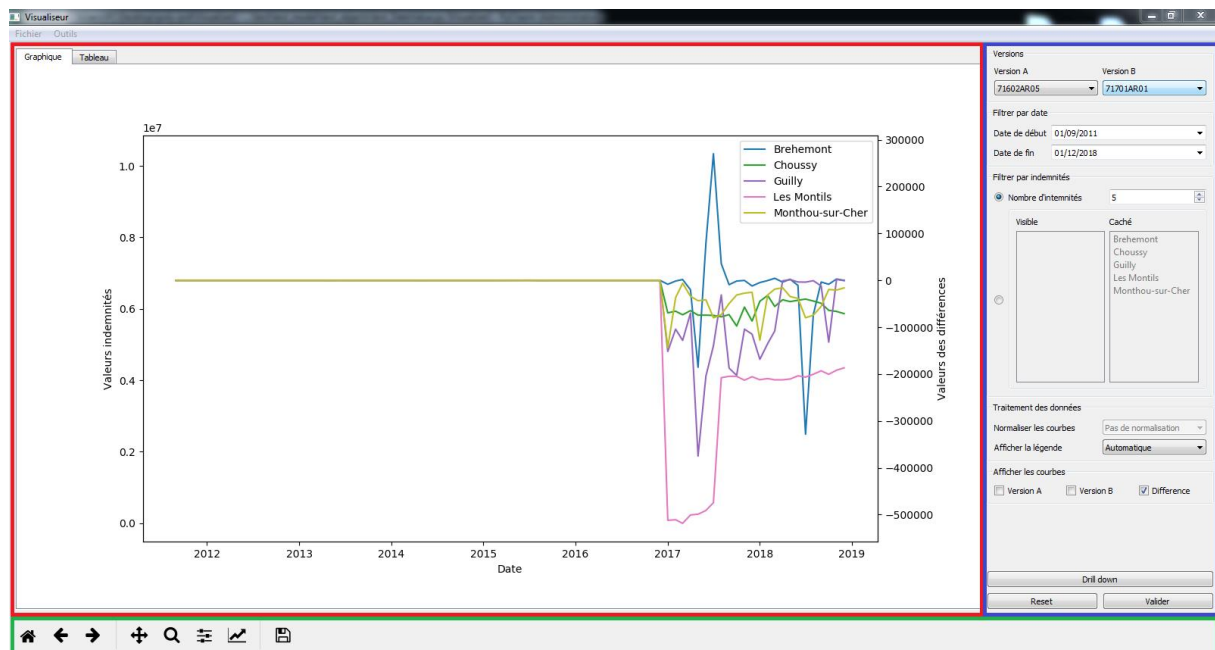


Illustration 25 : Capture d'écran de la fenêtre principale

On peut voir que la fenêtre se décompose en trois parties :

- La partie en **rouge** est la partie visualisation de l'application. C'est dans cette partie que sont affichées les courbes représentant l'évolution des indemnités dans le temps ou encore le tableau des données (explication plus détaillée dans la suite du rapport).
- La partie en **bleu** est la partie de filtrage des données.

- La partie en **verte** permet de manipuler le graphique. Cette barre d'outils permet de zoomer ou se déplacer dans le graphique. Elle permet aussi d'exporter la visualisation en image (au format matriciel ou vectoriel) ou en PDF.

### 5.2.2.1. Partie visualisation

La partie visualisation est la partie en **rouge** de l'illustration 25. Elle se décompose en 2 onglets : l'onglet graphique et l'onglet tableau de données.

L'onglet graphique permet de visualiser l'évolution des valeurs des indemnités dans le temps. La visualisation est décomposable en deux parties : visualisation d'une seule version de LOUVOIS ou visualisation de deux versions de LOUVOIS.

La visualisation d'une seule version de LOUVOIS étant très similaire à la visualisation de deux versions de LOUVOIS, je ne vais présenter que la visualisation avec deux versions.

Lorsqu'on visualise deux versions, le graphique fait apparaître 3 types de courbes. On a tout d'abord la courbe principale qui est la courbe des différences entre les deux versions du logiciel. Cette courbe est représentée en trait continu. On a ensuite une courbe représentant la version A du logiciel LOUVOIS en pointillés forts et une dernière courbe pour la version B en pointillés légers.

L'illustration ci-dessous représente dans l'ordre ces trois courbes.

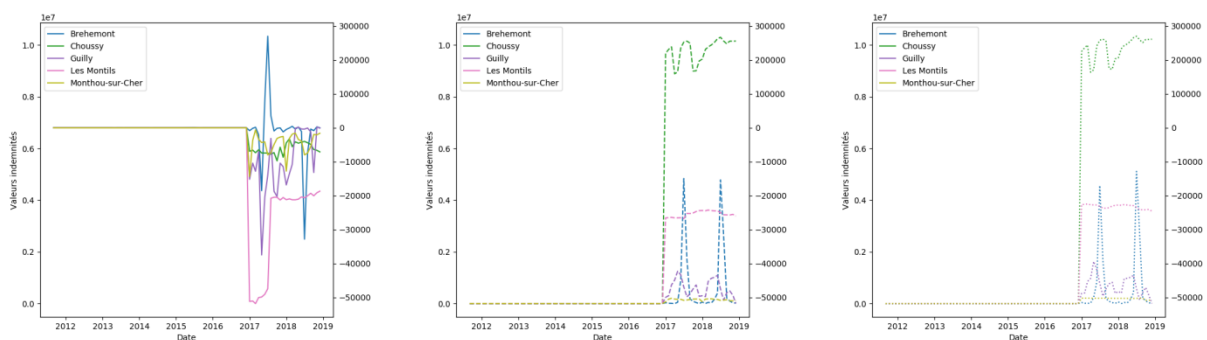


Illustration 26 : Trois types de courbes

Le graphique permet aussi d'activer ou de désactiver une indemnité avec un simple clic sur le trait de l'indemnité dans la légende. Cela permet de masquer certaines données sans à avoir à changer le réglage des filtres. Cela est aussi pratique pour épurer la visualisation lorsque celle-ci est trop encombrée par les différentes courbes.

Dans l'illustration ci-contre, toutes les indemnités sont désactivées excepté la rose. On y retrouve donc les 3 courbes cités précédemment.

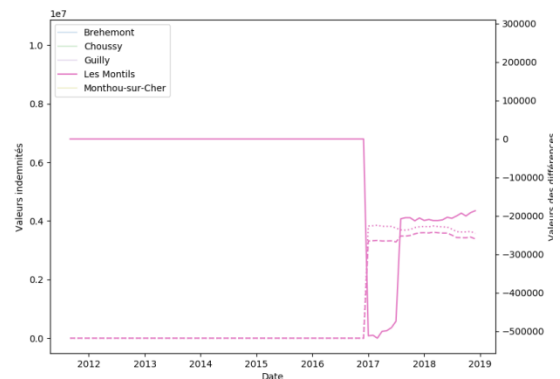


Illustration 27 : Visualisation avec des indemnités désactivées

Le deuxième onglet de la partie visualisation contient un tableau permettant à l'utilisateur de connaître les données qui sont utilisées pour générer le graphique.

Le tableau est décomposé en 6 colonnes : la date, le nom de l'indemnité, la valeur de l'indemnité de la version A, la valeur pour la version B, la différence entre les deux valeurs, ainsi que la différence absolue.

	Date	Indemnité	Valeur version A	Valeur version B	Différence	Différence absolue
1	2017/03/01	Les Montils	3333570.60	3852153.17	-518582.57	518582.57
2	2017/01/01	Les Montils	3314107.32	3826269.72	-512162.40	512162.40
3	2017/02/01	Les Montils	3324774.83	3835633.54	-510858.71	510858.71
4	2017/04/01	Les Montils	3316019.61	3816655.09	-500635.48	500635.48
5	2017/05/01	Les Montils	3316387.60	3815531.82	-499144.22	499144.22
6	2017/06/01	Les Montils	3320799.28	3811968.69	-491169.41	491169.41
7	2017/07/01	Les Montils	3282530.40	3756993.50	-474463.10	474463.10
8	2017/05/01	Guilly	1251531.21	1626528.08	-374996.87	374996.87
9	2018/07/01	Brehemont	4782311.81	5110748.82	-328437.01	328437.01
10	2017/07/01	Brehemont	4834089.51	4563838.84	270250.67	270250.67
11	2017/11/01	Les Montils	3559227.13	3772276.62	-213049.49	213049.49
12	2018/04/01	Les Montils	3596675.06	3808971.56	-212296.50	212296.50

Illustration 28 : Tableau de données

Il est possible de voir sur l'illustration ci-dessus que certaines cellules du tableau sont colorées. La couleur a en effet une signification.

- L'intensité de la couleur représente l'importance de la valeur de la cellule par rapport aux autres cellules de la même colonne. Si la valeur de la cellule est la valeur maximum de la colonne, alors l'intensité sera à son plus fort niveau.
- La couleur, orange ou rouge, représente le signe de la valeur. Si la valeur est négative, alors la cellule sera colorisée en orange, si elle est positive alors elle sera colorisée en rouge.

Aussi, l'application permet de filtrer les colonnes par ordre croissant ou décroissant sur un simple clic sur l'entête de la colonne.

### 5.2.2.2. Partie filtrage

La partie filtrage est la partie en **bleu** de l'illustration 25. Elle permet de filtrer les données que l'on souhaite visualiser.

La partie supérieure de la zone de filtrage permet de sélectionner quelles versions du logiciel LOUVOIS on souhaite visualiser. Dans le cas où la sélection de la version A correspond à celle de la version B, alors la visualisation se fera que sur une seule version. Sinon, la visualisation se fera avec les deux versions en se basant sur la différence entre les deux versions.

Elle permet aussi de sélectionner une date de début ainsi qu'une date de fin. Cela permet de réduire la fenêtre de temps visualisée et ainsi réduire la quantité de données à traiter par l'application.

Viens ensuite la sélection des indemnités que l'on souhaite visualiser. Se pose alors deux possibilités :

- Demander à l'application de nous fournir les X indemnités qui présentent les plus grosses différences.
- Sélectionner manuellement des indemnités spécifiques.

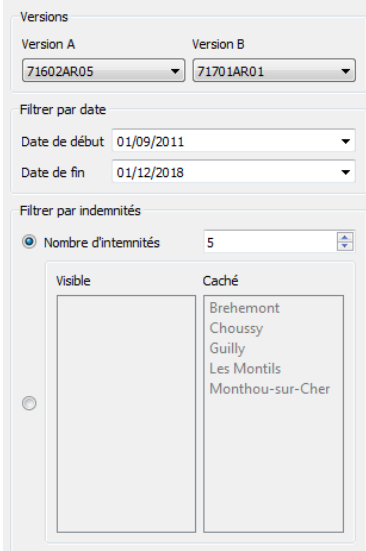


Illustration 29 : Partie supérieure de la zone de filtrage

La partie inférieure de la zone de filtrage permet de personnaliser l'affichage des données. L'utilisateur a donc la possibilité de forcer l'affichage ou non de la légende, ou bien de la laisser en automatique. Il lui est aussi possible d'afficher ou non un type spécifique de courbe : version A, version B, différence entre les deux versions.

Cette zone contient aussi 3 boutons.

- Le bouton « reset » permet de réinitialiser les valeurs de la zone de filtrage.
- Le bouton « valider » permet de valider les choix de filtrage et de mettre à jour la visualisation en fonction de ceux-ci.
- Le bouton « drill down » permet d'accéder à la seconde fenêtre permettant de visualiser la répartition spatiale des individus. La configuration actuelle des filtres est transmise automatiquement à cette fenêtre.

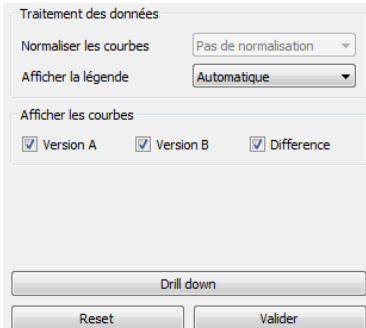


Illustration 30 : Partie inférieure de la zone de filtrage

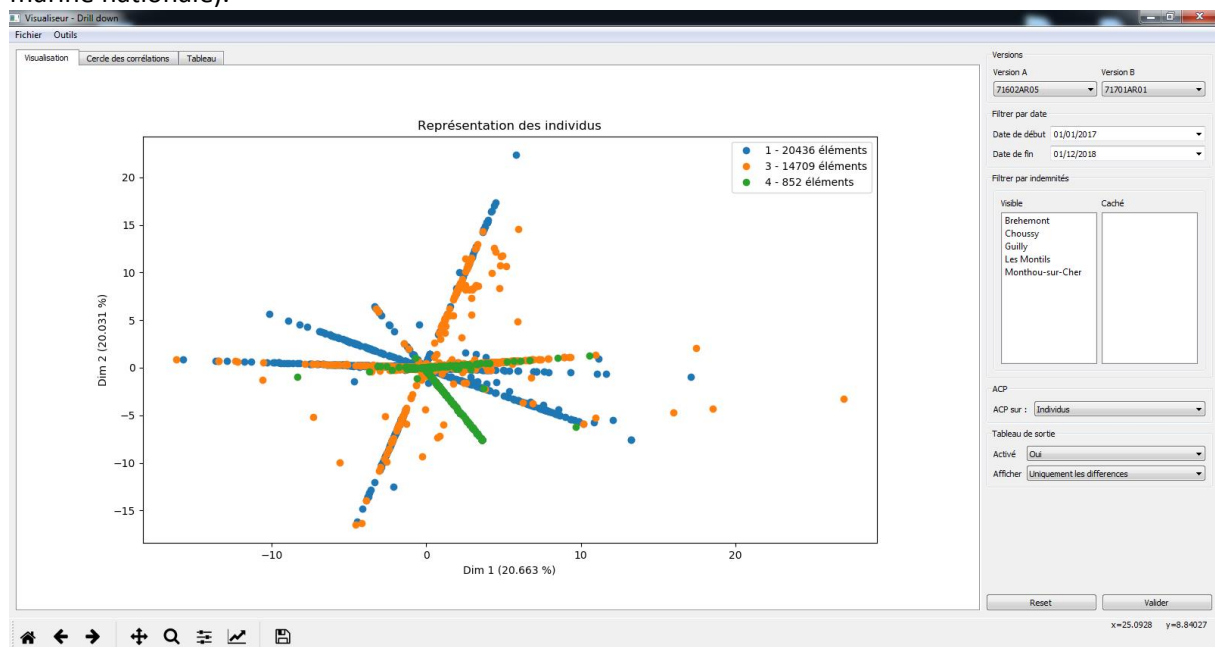
### 5.2.3. Fenêtre secondaire : exploration des individus

Cette fenêtre permet de visualiser la répartition spatiale des individus dans l'espace. Elle permet aussi de visualiser quelles sont les indemnités qui se ressemblent.

Cette fenêtre se décompose en trois parties comme pour la fenêtre principale. La zone de filtrage reste sensiblement la même. En revanche les visualisations proposées sont totalement différentes.

Cette fenêtre propose donc une visualisation de la répartition des individus dans l'espace, une visualisation de la corrélation des indemnités, un tableau de données (comme pour la fenêtre principale).

Voici un exemple de visualisation des individus dans l'illustration ci-dessous. Chaque point (individu) est coloré en fonction de son appartenance à un corps d'armée (armée de terre, armée de l'air, marine nationale).

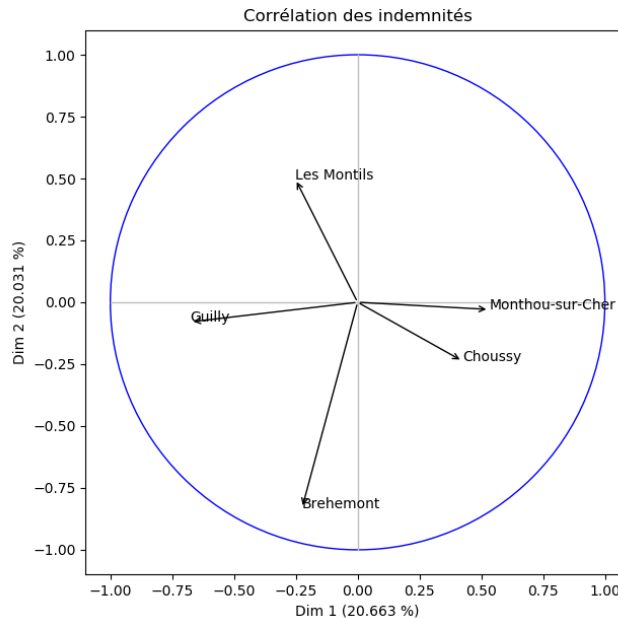


**Illustration 31 : Représentation spatiale des individus**

La représentation spatiale des individus pose un problème. En effet, l'utilisateur peut choisir un nombre quelconque d'indemnités. Chaque indemnité représente une dimension. Un individu se situe en fonction de ses valeurs pour chaque dimension. Le problème est que l'on représente les données en 2 dimensions. Il faut donc un moyen de convertir toutes ces dimensions en seulement 2 dimensions.

Pour réaliser cela, j'ai utilisé une transformation mathématique qui se nomme ACP (analyse en composantes principales). Cette transformation mathématique va chercher à déterminer un repère qui permet de représenter au mieux les données. Une fois ce repère déterminé, il ne reste plus qu'à projeter les points sur le nouveau repère. Le problème de ce nouveau repère est que les axes n'ont plus de réelles significations. En effet, chaque axe est issu d'une composante de plusieurs indemnités.

Afin d'essayer d'interpréter la signification de chaque axe, on peut utiliser une représentation graphique intitulé le cercle des corrélations (voir illustration ci-après). Ce cercle permet de mettre en évidence les indemnités qui participent à l'existence de l'axe.



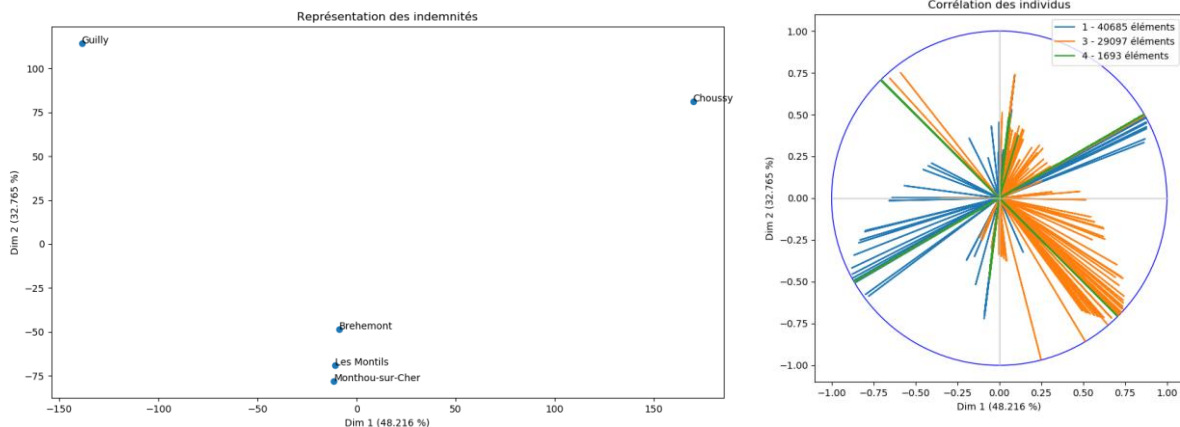
**Illustration 32 : Cercle des corrélations**

Le cercle des corrélations permet aussi de savoir les indemnités qui se ressemblent. En effet, plus elles sont proches, plus cela signifie qu'elles se ressemblent. Par exemple, on peut dire que « Choussy » et « Monthou-sur-Cher » se ressemblent mais n'ont rien à voir avec « Les Montils ».

Cependant, l'interprétation des graphiques est à faire avec prudence. En effet, à côté de chaque nom de dimension, est noté le pourcentage de représentativité (inertie) des données initiales. On peut voir que dans l'exemple précédent, l'inertie représentée n'est que de 40% ce qui signifie qu'il y a 60% des données qui sont déformées, et donc qui peuvent potentiellement erronées notre interprétation.

Le troisième onglet contient le tableau des données utilisées pour réaliser les représentations graphiques. Il fonctionne sur le même principe que pour la fenêtre principale, je ne vais donc pas représenter de nouveau son fonctionnement.

Une option de filtrage permet d'inverser les données visualisées. Au lieu de visualiser la répartition des individus, on peut visualiser la répartition des indemnités dans l'espace. On va donc aussi pouvoir visualiser les différentes corrélations entre les individus.



**Illustration 33: Représentation spatiale des indemnités**

Sur cette fenêtre, il faut faire particulièrement attention aux données renseignées dans la zone de filtrage. En effet, cette visualisation manipule beaucoup de données puisqu'elle travaille sur les individus. Il est donc impératif que l'utilisateur fasse attention à bien choisir une zone de visualisation restreinte afin d'éviter une trop forte consommation de mémoire RAM et d'allonger le temps d'exécution.

### 5.3. Structure de l'application

Je vais décrire dans cette section la structure de l'application. L'objectif est de comprendre le découpage de l'application pour ainsi faciliter la prise en main de l'application d'un point de vue développement.

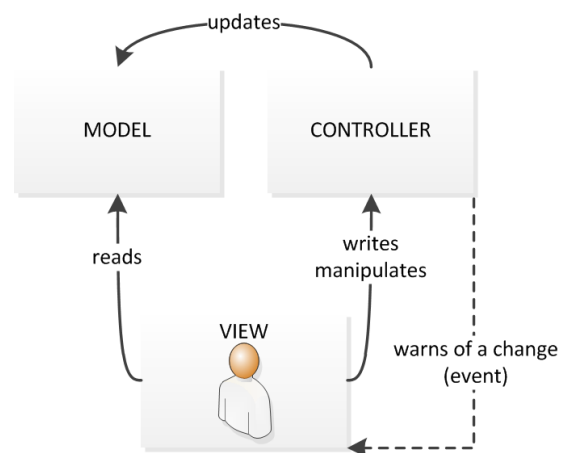
#### 5.3.1. Pattern MVC

Afin de structurer l'application, j'ai choisi de découper le code suivant le pattern MVC (Modèle, Vue, Contrôleur). Le but premier de suivre une architecture particulière est de s'imposer une norme pour que le code reste structuré et compréhensible.

L'objectif du pattern MVC est de séparer le rôle de chaque classe afin que celles-ci aient un but et un seul but spécifique à remplir.

Le rôle de chaque partie est de :

- Le modèle contient les données à afficher.
- La vue contient la présentation de l'interface graphique.
- Le contrôleur contient la logique concernant les actions effectuées par l'utilisateur.



**Illustration 34 : Schéma de fonctionnement du pattern MVC**

### 5.3.2. Diagramme UML

Ci-dessous se trouve le diagramme UML des classes de l'application. On voit rapidement le pattern MVC apparaitre : en vert les classes des vues, en bleu les classes des contrôleurs, et les classes sur fond blanc font parties du modèle.

J'ai volontairement masqué les méthodes des classes des vues, des contrôleurs, ainsi que celles des classes liées à la base de données pour des raisons de lisibilité. En effet, ces classes comportent de nombreuses méthodes et ce n'est pas vraiment pertinent de les afficher ici.

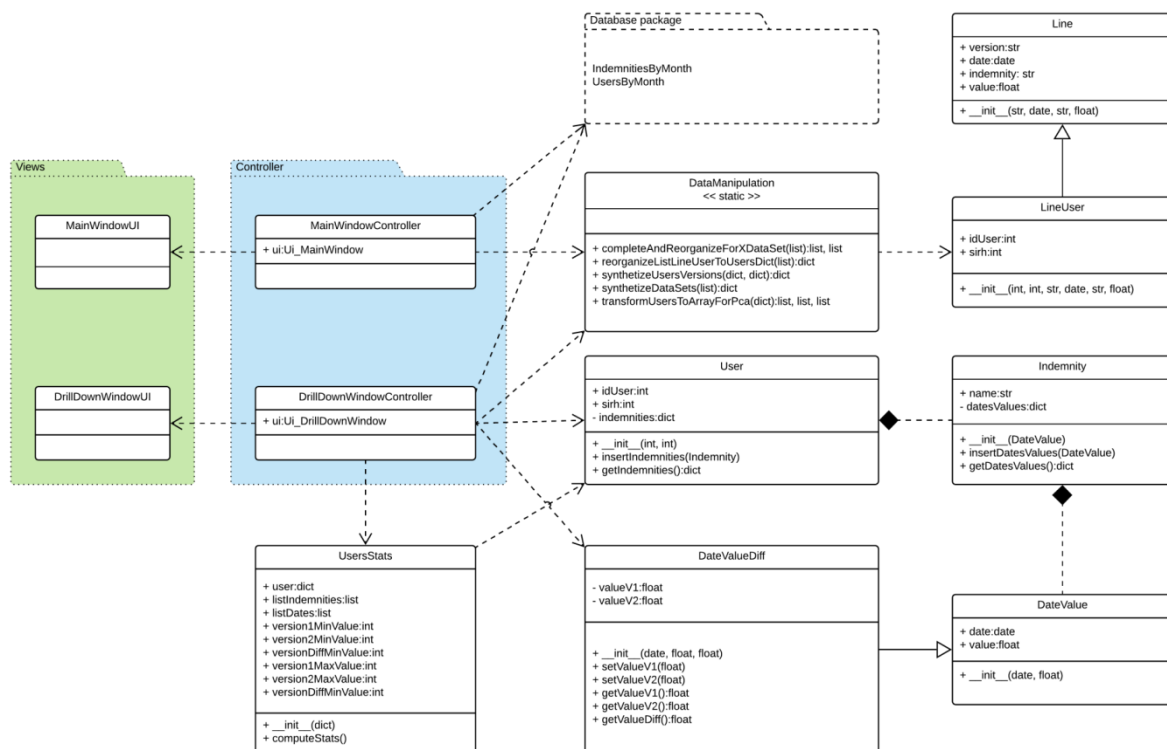


Illustration 35 : Diagramme UML de l'application

## 5.4. Spécifications machine

Cette section détaille les spécifications machine pour faire fonctionner l'application. Les valeurs sont des indications, il se peut que selon la machine utilisée, cela diffère.

L'application fonctionne dans un environnement Windows 7 – 64 bits.

La fenêtre principale ne demande pas beaucoup de performance par rapport à la seconde. Donc les chiffres sont basés sur la seconde fenêtre.



L'application est mono-thread et n'est pas très bien optimisée niveau mémoire vive. Il y a donc des possibilités d'amélioration de ce côté. A cause de cela, un fort pic de mémoire est utilisé pendant l'exécution.

Les résultats sont donnés pour une visualisation des individus pour 5 indemnités sur un intervalle de temps de 2 ans (environ 40 000 individus).

	Temps	Mémoire utilisé	Mémoire utilisé en pic
<b>Avec le tableau de sortie désactivé</b>	1 minute 10 secondes	123 Mo	880 Mo
<b>Avec le tableau de sortie activé</b>	2 minutes 30 secondes	1 Go	1.5 Go

## 5.5. Les difficultés de la réalisation

L'application ne paraît pas très complexe au premier abord. Cependant, durant son développement, j'ai rencontré de nombreux problèmes notamment liés à la consommation de mémoire RAM.

En effet, l'application permet à l'utilisateur de travailler sur une grande quantité de données notamment lorsqu'il souhaite consulter la répartition spatiale des individus. La grande quantité de données manipulées impose de réaliser des optimisations afin que l'application soit à la fois performante et la moins consommatrice possible de mémoire vive.

Cette contrainte de consommation de mémoire a été très importante car elle m'a imposé de repenser plusieurs fois la façon de stocker et de manipuler les données. J'ai donc dû faire des compromis entre une conception orientée objet et une conception de stockage dans de simples listes / tableau (ce qui réduit la facilité de compréhension du code).

Aussi, avant de pouvoir manipuler les données dans l'application, j'ai dû faire des requêtes en base de données pour rapatrier les données en fonction des réglages des filtres. Cependant, la base de données est conséquente et donc faire une requête peut demander un certain temps de traitement. Il a donc fallu que je limite le nombre de requêtes faites à la base de données. C'est notamment pour cela que j'ai mis en place la possibilité d'activer/désactiver une indemnité sur le graphique de la fenêtre principale afin de réduire le nombre de requête sur la base de données.

La bibliothèque graphique a aussi montré à plusieurs reprises ses limites. Il a donc fallu l'utiliser d'une autre façon ou alors user de subterfuge pour pouvoir arriver au résultat souhaité.

## 6. Démarche Qualité

La démarche qualité a été un point très important sur lequel j'ai accordé beaucoup de ressources. En effet, l'objectif de l'application est de déceler des anomalies d'une version à l'autre de LOUVOIS et d'en comprendre les origines. Il faut donc que l'application permettant de faire cela soit fiable.

La démarche qualité a aussi pour but de faciliter la reprise et la maintenabilité du code. Aussi, un code de qualité est moins assujéti à se comporter de façon anormale, et donc de provoquer des bugs.

### 6.1. Les bibliothèques

L'application a été développée en Python. Afin de faciliter le développement et de gagner du temps, j'ai utilisé différentes bibliothèques : Matplotlib, Scikit-learn, Qt.

#### 6.1.1. Matplotlib

La bibliothèque Matplotlib permet de tracer des graphiques. C'est une bibliothèque très puissante et elle est capable de gérer une grande quantité de données. Elle possède aussi un ensemble d'outils (comme le zoom par exemple) permettant de manipuler les graphiques. Elle offre aussi la possibilité d'exporter la visualisation en différents formats comme en images PNG, JPG, SVG ou encore en PDF.

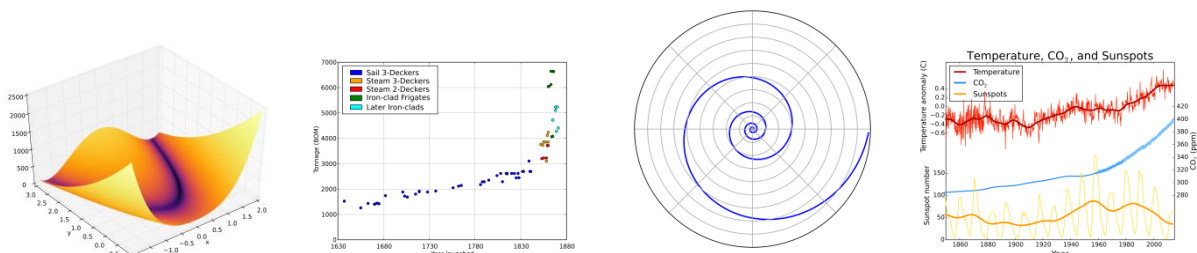


Illustration 36 : Exemples de représentations avec Matplotlib

#### 6.1.2. Scikit-learn

Scikit-learn est une bibliothèque de traitement de données. Elle permet estimer des forêts aléatoires, des régressions logistiques, des algorithmes de classification, et les machines à vecteurs de support. Elle est destinée à l'apprentissage automatique.

J'utilise cette bibliothèque pour réaliser l'Analyse en Composante Principales (ACP).

#### 6.1.3. Qt

Pour réaliser les interfaces graphiques, j'ai utilisé la bibliothèque PySide et PyQt. Ces bibliothèques sont dérivées du projet Qt initialement développé en C++. Elles permettent de fournir une alternative

aux interfaces Tkinter qui ne sont pas très belle. Elles permettent aussi d'ajouter de nombreux éléments graphiques pour faciliter le développement.

Aussi, PyQt met à disposition un outil, QtDesigner, permettant de concevoir graphiquement ses interfaces. Cela permet de construire la fenêtre avec des glissés/déposés d'éléments comme des boutons par exemple. Cela apporte une facilité de développement.

Ensuite, un second outil permet de convertir la modélisation de la fenêtre en une classe Python.

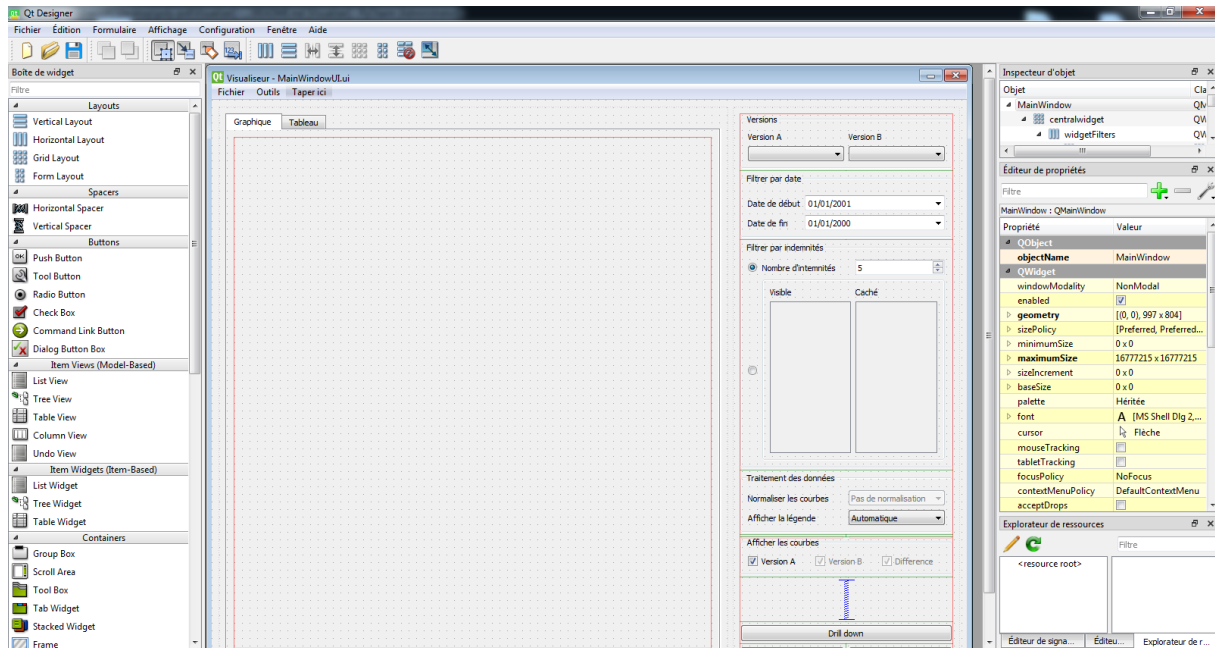


Illustration 37 : Aperçu de QtDesigner avec la fenêtre principale en édition

## 6.2. Tests

Afin de vérifier la qualité et la fiabilité des fonctions développées, j'ai mis en place des tests unitaires. J'ai effectué les tests unitaires sur toutes les fonctions manipulant les données.

Les tests unitaires permettent de vérifier le bon fonctionnement d'une fonction, et qu'il n'y a pas de bug qui apparaît durant la suite du développement. Ainsi, tester la fonction se fait de manière automatique et rapide.

La librairie « nose » permet de lancer l'ensemble des tests unitaires de l'application puis de faire un rapport. Cela permet en une commande de tester toute l'application de manière automatique.

```
test_dataValue (test_DateValue.test_DataManipulation) ... ok
test_dataValueDiff (test_DateValueDiff.test_DataManipulation) ... ok
test_dataValueDiffChangeV1 (test_DateValueDiff.test_DataManipulation) ... ok
test_dataValueDiffChangeV2 (test_DateValueDiff.test_DataManipulation) ... ok
test_Indemnity (test_Indemnity.test_Indemnity) ... ok
test_Line (test_Line.test_Line) ... ok
```

-----  
Ran 6 tests in 0.204s

OK

Illustration 38 : Résultat d'exécution de nose

## 6.3. GitLab

Durant le développement, j'ai utilisé la plateforme GitLab. GitLab est une plateforme proposant divers services et repose le fonctionne du logiciel de versionning Git.

### 6.3.1. Gestion des versions

J'ai donc utilisé GitLab afin de gérer les versions de mon code. Chaque modification (ajout d'une fonctionnalité, correction d'un bug, ...) a été versionné indépendamment les unes des autres. Faire cela permet de garder un historique clair et précis de modifications apportées au code. Cela permet aussi de pouvoir revenir en arrière en cas de problème. Par exemple, certaines implémentations ont été mal conçues et consommées trop de RAM, par conséquent, je suis revenu a une version plus fonctionnelle de mon application pour réaliser une implémentation différente qui été mieux optimisé.

### 6.3.2. Tickets

J'ai aussi utilisé GitLab pour gérer les tickets (issues). J'ai beaucoup utilisé les tickets pour gérer une liste des choses à faire et des choses faites.

Comme le montre la capture d'écran ci-contre, chaque ticket possédait un ou plusieurs tags de couleurs différentes. Cela m'a permit de différencier les taches de correction de bugs, ou les tahes d'ajout de fonctionnalités.

La liste a évolué au court du temps en fonction des bugs découvert, des fonctionnalités qui ont été ajoutées, des tickets qui ont été effectués, ...

[Visualizer] - Drill down function	To Do
#5 · opened 4 weeks ago by Guillaume Servais	
[Visualizer] X-axis don't appear in some cases	bug
#7 · opened 3 weeks ago by Guillaume Servais	
[Visualizer] Drill down : clear all view BEFORE execute request (optimize RAM)	To Do
#8 · opened 1 week ago by Guillaume Servais	
[Visualizer] Legend lines don't appear	bug
#2 · opened 4 weeks ago by Guillaume Servais	
[Visualizer] - Make a filter on differences between indemnities	To Do
#4 · opened 4 weeks ago by Guillaume Servais	
[Visualizer] - Show data in tab pane	To Do
#3 · opened 4 weeks ago by Guillaume Servais	
[Visualizer] Legend don't appear when filter indemnities by name was selected	bug
#6 · opened 3 weeks ago by Guillaume Servais	
[Visualizer] Legend click event not work everytime	bug
#1 · opened 4 weeks ago by Guillaume Servais	

Illustration 39 : Aperçu de la liste des tickets

### 6.3.3. Intégration continue

Enfin, la troisième et dernière fonctionnalité que GitLab que j'ai utilisé est l'intégration continue. J'ai utilisé l'intégration continue pour automatiser certaines tâches.

A chaque nouvelle version de l'application, GitLab exécute les tests unitaires et génère un rapport. Si les tests unitaires se sont bien déroulés, alors une analyse de code est effectuée via l'outil SonarQube. Cette automatisation permet de détecter rapidement une anomalie dans le code. En effet, si les tests échouent, GitLab nous prévient par mail que les tests ne se sont pas déroulés correctement. On peut donc corriger rapidement le problème.

Dans l'illustration ci-dessous, on peut voir deux bulles vertes qui représentent les deux tâches qui s'exécutent à chaque version de code.

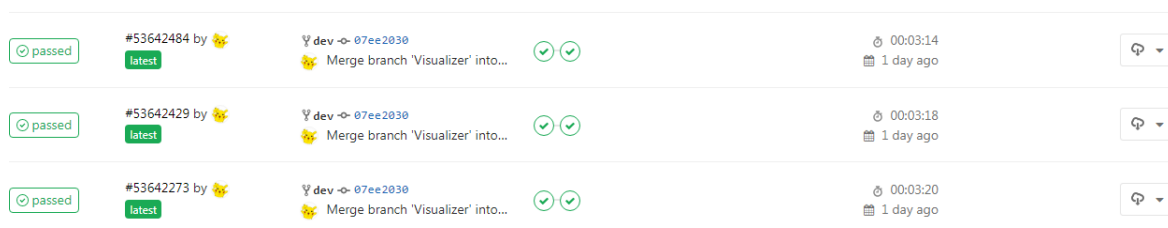


Illustration 40 : GitLab - Intégration continue

## 6.4. SonarQube

SonarQube est un outil de mesure en continu de la qualité de code. Il est capable d'effectuer une analyse sur plus de 25 langages différents.

SonarQube permet de calculer de nombreuses statistiques comme la duplication de code, mesure du niveau de documentation, le respect des règles de programmation, l'identification de bugs et de vulnérabilités, la couverture du code par les tests unitaires, la complexité des fonctions, ... Il est aussi possible d'ajouter d'autres statistiques en lui ajoutant des plugins.

A partir des résultats statistiques, SonarQube est capable d'estimer la quantité de travail en temps Homme pour corriger tous les défauts détectés. Cela en fait donc un outil extrêmement puissant.

L'illustration 41 ci-dessous, est le rapport de SonarQube pour le projet. On y voit qu'il n'y a pas de bugs détectés, ni de duplication de code. En revanche, on peut voir que les tests unitaires ne couvrent que 14% de la totalité du code. Cela s'explique par le fait que je n'ai testé que les fonctions manipulant les données. Les autres fonctions sont des fonctions de lectures dans la base de données, ou d'affichage graphique des données. Je n'ai pas effectué ces tests car ils présentent des particularités techniques qui me demande un temps de formation or, j'ai manqué de temps pour réaliser cela.

Cependant, je peux affirmer que 100% des méthodes de manipulation des données ont été testés par des tests unitaires.

Aussi SonarQube indique 2 jours de temps humain pour corriger les problèmes qu'il a détecté. Ces problèmes n'ont pas une criticité élevée. Ils sont provoqués en grande partie parce que je n'ai pas respecté la convention de nommage du Python. En effet, j'ai respecté la convention de nommage CamelCase alors que j'aurai dû utiliser la convention snake\_case. Cependant, je m'en suis aperçu un peu tard et j'ai préféré continuer de respecter ma convention initiale plutôt que de mélanger les deux conventions.

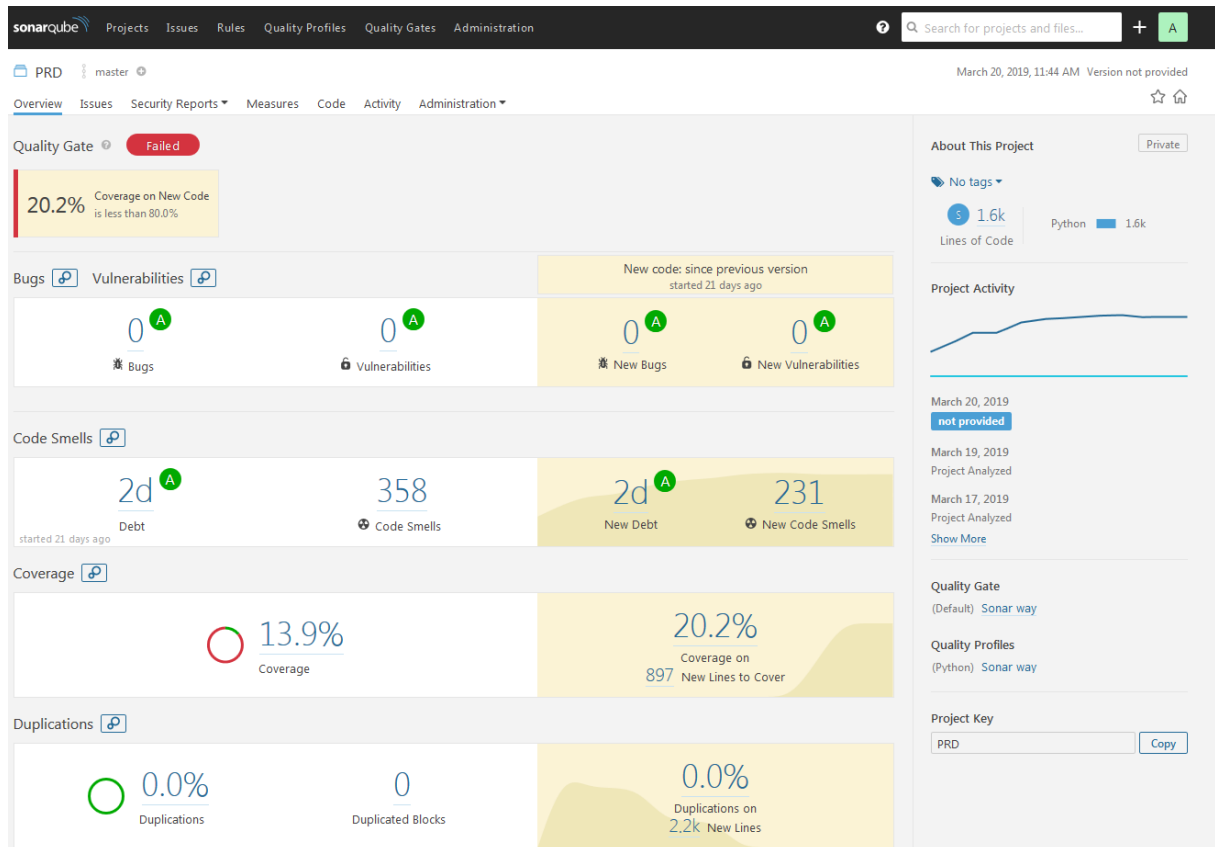


Illustration 41 : Rapport SonarQube

## Bilan et conclusion

Ce projet de recherche et développement en partenariat avec la société Sopra Steria est un projet légèrement plus orienté recherche que développement. Cependant une grosse partie de développement a été réalisée.

Ce projet a nécessité beaucoup de temps de compréhension du sujet. Il a aussi nécessité beaucoup de recherche de l'état de l'art pour connaître les moyens utilisés pour répondre à la problématique de recherche d'anomalie. Cette recherche a été relativement compliquée car l'analyse de données généralement pratiquée, s'intéresse davantage aux tendances globales des données tandis que ce projet se concentre sur l'inverse.

Durant le développement, j'ai été régulièrement confronté à des limitations techniques auxquelles j'ai dû faire face avec plus ou moins de difficultés.

Le projet a été difficile à gérer tellement l'objectif initial était vaste. Cependant, tout au long du projet, l'objectif a été revu à la baisse et régulièrement ajusté. Par rapport au planning fourni lors du rapport de mis projet (voir annexe), il y a eu un retard sur la phase de développement de la partie visualisation. Ce retard a été compensé par une diminution de la quantité de l'analyse de données qui était prévue initialement.

Le projet a été l'occasion de faire une phase d'état de l'art permettant d'explorer les différentes techniques et possibilités qui existaient pour faire face à la problématique. Ce projet a ensuite permis de reproduire l'outil utilisé par Sopra Steria et de lui ajouter des fonctionnalités. Il permet donc de visualiser l'évolution des valeurs des indemnités dans le temps, puis de sélectionner un lot de données pour en explorer les individus qui composaient ce lot de données.

Malheureusement, l'exploration des individus n'apporte pour le moment pas beaucoup d'information. En effet, à ce stade, il manque des critères de caractérisation des individus pour pouvoir déterminer les facteurs à l'origine des anomalies détectées. Il faut donc que l'application soit étendue pour devenir un outil indispensable à l'entreprise.

Enfin je terminerai ce rapport par dire que je suis assez fier de ce que j'ai réalisé. En effet, le début du projet m'avait complètement perdu et je ne me voyais pas produire quelque chose. Au final le rendu est plutôt satisfaisant de mon point de vue.

Un élément que je tire de ce projet est qu'il a été difficile de travailler seul sur un tel projet. J'avais régulièrement des choix techniques à faire et avoir un autre avis aurait peut-être pu me permettre de prendre d'autres directions plus simple à mettre en œuvre.

## Table des illustrations

Illustration 1 : Schéma du fonctionnement de la méthode d'analyse existante .....	11
Illustration 2 : Structure générale du système .....	13
Illustration 3 : Représentation en nuage de points .....	14
Illustration 4 : Représentation radiale .....	15
Illustration 5 : Evolution du nombre d'objets connectés .....	16
Illustration 6 : Carte thermique .....	17
Illustration 7 : Simplification d'un nuage de points .....	17
Illustration 8 : Graphe en 2 et 3 dimensions .....	18
Illustration 9 : Représentation d'un individu .....	18
Illustration 10 : Exemple de rendu avec une représentation basé sur les pixels .....	19
Illustration 11 : Arbre hiérarchique .....	19
Illustration 12 : Arbre hyperbolique .....	20
Illustration 13 : Schéma de classification des algorithmes d'analyse de données .....	21
Illustration 14 : Evolutions du calcul du K-Means .....	22
Illustration 15 : Limitation du K-Means .....	22
Illustration 16 : Clustering hiérarchique .....	23
Illustration 17 : DBSCAN détection du bruit .....	24
Illustration 18 : DBSCAN détection des classes .....	24
Illustration 19 : Comparaison K-Means - DBSCAN .....	25
Illustration 20 : Représentation d'anomalies ponctuelles .....	26
Illustration 21 : Représentation d'une anomalie contextuelle .....	26
Illustration 22 : Représentation d'une anomalie collective .....	27
Illustration 23 : Représentation de données bruitées .....	28
Illustration 24 : Séquence d'utilisation de l'application .....	33
Illustration 25 : Capture d'écran de la fenêtre principale .....	33
Illustration 26 : Trois types de courbes .....	34
Illustration 27 : Visualisation avec des indemnités désactivées .....	34
Illustration 28 : Tableau de données .....	35
Illustration 29 : Partie supérieure de la zone de filtrage .....	36
Illustration 30 : Partie inférieure de la zone de filtrage .....	36
Illustration 31 : Représentation spatiale des individus .....	37
Illustration 32 : Cercle des corrélations .....	38
Illustration 33 : Représentation spatiale des indemnités .....	39
Illustration 34 : Schéma de fonctionnement du pattern MVC .....	39
Illustration 35 : Diagramme UML de l'application .....	40
Illustration 36 : Exemples de représentations avec Matplotlib .....	42
Illustration 37 : Aperçu de QtDesigner avec la fenêtre principale en édition .....	43
Illustration 38 : Résultat d'exécution de nose .....	44
Illustration 39 : Aperçu de la liste des tickets .....	44
Illustration 40 : GitLab - Intégration continue .....	45
Illustration 41 : Rapport SonarQube .....	46
Illustration 42 : Maquette de l'interface homme/machine .....	53



## Bibliographie

1. **Yannis Chaouche, Chloé-Agathe Azencott.** Découvrez l'intérêt des algorithmes de clustering. *openclassrooms.com*. [En ligne] [Citation : 17 octobre 2018.] <https://openclassrooms.com/fr/courses/4379436-explorez-vos-donnees-avec-des-algorithmes-non-supervises/4379551-decouvrez-l-interet-des-algorithmes-de-clustering>.
2. **Weber, Wolfgang.** Les 5 plus grandes difficultés de la détection d'anomalies. *leanbi.ch*. [En ligne] 10 février 2017. [Citation : 20 septembre 2018.] <https://leanbi.ch/fr/blog/5-grandes-difficultes-de-la-detection-danomalies/>.
3. **Tianyang Liu, Fatma Bouali, Gilles Venturini.** On visualizing large multidimensional datasets with a multi-threaded radial approach. [En ligne] mars 2015. [Citation : 20 septembre 2018.] [https://www.researchgate.net/publication/273477530\\_On\\_visualizing\\_large\\_multidimensional\\_data\\_sets\\_with\\_a\\_multi-threaded\\_radial\\_approach](https://www.researchgate.net/publication/273477530_On_visualizing_large_multidimensional_data_sets_with_a_multi-threaded_radial_approach).
4. **Raval, Siraj.** *The Best Way to Visualize a Dataset Easily*. [En ligne] 23 décembre 2016. [Citation : 20 septembre 2018.] <https://www.youtube.com/watch?v=yQsOFWqpjKE>.
5. **L, Bastien.** Analyse de données : top 5 des algorithmes Big Data Analytics. [En ligne] 03 juillet 2018. [Citation : 22 novembre 2018.] <https://www.lebigdata.fr/analyse-de-donnees-algorithmes>.
6. **Keim, Daniel A.** Visual Data Mining. [En ligne] septembre 1997. [Citation : 07 novembre 2018.] <http://kops.uni-konstanz.de/handle/123456789/5713>.
7. **Johnson, Chris.** Visualizing large data sets: Chris Johnson at TEDxSaltLakeCity. [En ligne] 20 juin 2011. [Citation : 20 septembre 2018.] <https://www.youtube.com/watch?v=5UxC9Le1eOY>.
8. **Hammami, Donia.** Techniques du data mining. [En ligne] 11 mai 2016. [Citation : 03 octobre 2018.] <https://fr.slideshare.net/doniahammami/techniques-du-data-mining>.
9. **Fieschi, Marius.** Data Mining, fouille de données: Concepts et techniques. [En ligne] février 2006. [Citation : 03 octobre 2018.] [http://cybertim.timone.univ-mrs.fr/enseignement/doc-enseignement/informatique/introdatawarehouse/docpeda\\_fichier](http://cybertim.timone.univ-mrs.fr/enseignement/doc-enseignement/informatique/introdatawarehouse/docpeda_fichier).
10. **Christos Faloutsos, King-Ip Lin.** FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets. [En ligne] juin 1997. [Citation : 20 septembre 2018.] [https://www.researchgate.net/publication/2609830\\_FastMap\\_A\\_Fast\\_Algorithm\\_for\\_Indexing\\_Data-Mining\\_and\\_Visualization\\_of\\_Traditional\\_and\\_Multimedia\\_Datasets](https://www.researchgate.net/publication/2609830_FastMap_A_Fast_Algorithm_for_Indexing_Data-Mining_and_Visualization_of_Traditional_and_Multimedia_Datasets).
11. **Bellas, Anastasios.** Détection d'anomalies à la volée dans des flux de données de grande dimension. [En ligne] 10 février 2014. [Citation : 20 septembre 2018.] <https://tel.archives-ouvertes.fr/tel-00944263/document>.
12. Introduction au Data Mining et à l'apprentissage statistique. [En ligne] [Citation : 03 octobre 2018.] <http://cedric.cnam.fr/~saporta/DM.pdf>.
13. Heatmap des lieux de tournage de films à Paris. *data.gouv.fr*. [En ligne] 02 mars 2014. [Citation : 06 décembre 2018.] <https://www.data.gouv.fr/fr/reuses/heatmap-des-lieux-de-tournage-de-films-a-paris/>.
14. Data mining. [En ligne] [Citation : 03 octobre 2018.] [http://dspace.univ-biskra.dz:8080/jspui/bitstream/123456789/5242/10/I\\_Data%20Mining.pdf](http://dspace.univ-biskra.dz:8080/jspui/bitstream/123456789/5242/10/I_Data%20Mining.pdf).
15. Biclustering. *wikipedia.org*. [En ligne] [Citation : 03 octobre 2018.] <https://en.wikipedia.org/wiki/Biclustering>.
16. Embedding matplotlib in Qt. *matplotlib.org*. [En ligne] [https://matplotlib.org/gallery/user\\_interfaces/embedding\\_in\\_qt\\_sgskip.html](https://matplotlib.org/gallery/user_interfaces/embedding_in_qt_sgskip.html).
17. Matplotlib Legend Picking. *matplotlib.org*. [En ligne] [https://matplotlib.org/gallery/event\\_handling/legend\\_picking.html](https://matplotlib.org/gallery/event_handling/legend_picking.html).

- 
18. Les docstrings en Python. *Sam & Max*. [En ligne] <http://sametmax.com/les-docstrings/>.
19. **Rakotomalala, Ricco**. ACP (analyse en composantes principales) sous Python. *eric.univ-lyon2.fr*. [En ligne] 8 juin 2018. [http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/fr\\_Tanagra\\_ACP\\_Python.pdf](http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/fr_Tanagra_ACP_Python.pdf).

## Annexes

1.	Description des interfaces externes du logiciel.....	53
1.1.	Interfaces matériel/logiciel .....	53
1.2.	Interfaces homme/machine .....	53
1.3.	Interfaces logiciel/logiciel.....	53
2.	Spécifications fonctionnelles.....	54
2.1.	Définition de la fonction : Calculer une représentation graphique des données .....	54
2.2.	Définition de la fonction : Filtrer les données par caractéristiques des individus .....	54
2.3.	Définition de la fonction : Filtrer les données par indemnités.....	54
2.4.	Définition de la fonction : Points communs entre individus .....	55
2.5.	Définition de la fonction : Prédiction des individus potentiellement problématiques.....	55
2.6.	Définition de la fonction : Sauvegarde des résultats .....	55
3.	Spécifications non fonctionnelles.....	56
3.1.	Contraintes de développement et conception .....	56
3.2.	Contraintes de fonctionnement et d'exploitation .....	56
3.2.1.	Performances .....	56
3.2.2.	Capacités .....	56
3.2.3.	Contrôlabilité.....	56
3.2.4.	Sécurité.....	57
4.	Comptes rendus hebdomadaires (Weekly) .....	58
5.	Document utilisateur.....	68
5.1.	Introduction.....	68
5.2.	Installation.....	69
5.2.1.	Python .....	69
5.2.2.	Dépendances de l'application .....	69
5.2.3.	Configuration.....	70
5.3.	Utilisation .....	71
5.3.1.	Lancer l'application .....	71
5.3.2.	Décomposition de la fenêtre.....	72
5.3.3.	Utilisation de la fenêtre principale.....	72
5.3.4.	Utilisation de la fenêtre de drill down.....	75
6.	Document développeur.....	77
6.1.	Introduction.....	77
6.2.	Installation.....	78

---

6.2.1.	Python .....	78
6.2.2.	Environnement de développement (IDE).....	78
6.2.3.	Dépendances de l'application .....	79
6.2.4.	Base de données.....	80
6.2.5.	Configuration.....	80
6.3.	Architecture.....	81
6.4.	Modifier les vues .....	82
6.5.	Tests unitaires .....	83
7.	Rapport SonarQube.....	84
8.	Diagramme de Gantt prévisionnel .....	85
9.	Diagramme de Gantt réel.....	86

## 1. Description des interfaces externes du logiciel

### 1.1. Interfaces matériel/logiciel

Afin de diminuer les délais d'interactions entre chaque action effectuée par l'utilisateur, l'application parallélisera les calculs avec une carte graphique. L'interaction entre l'application et la carte graphique se fera via la librairie CUDA d'NVIDIA.

### 1.2. Interfaces homme/machine

L'interface homme/machine permettra à l'utilisateur de consulter l'analyse des données et d'explorer les points. Une fonctionnalité zoom/dé-zoom permettra d'agrandir certaines zones pour les consulter avec plus de détails. Aussi un volet sera consacré à l'édition de filtres pour affiner les résultats visibles.

L'application sera donc scindée en deux parties comme sur la maquette ci-dessous : la partie de gauche servant à consulter les données et la partie de droite servant à régler les filtres, changer la visualisation, ...

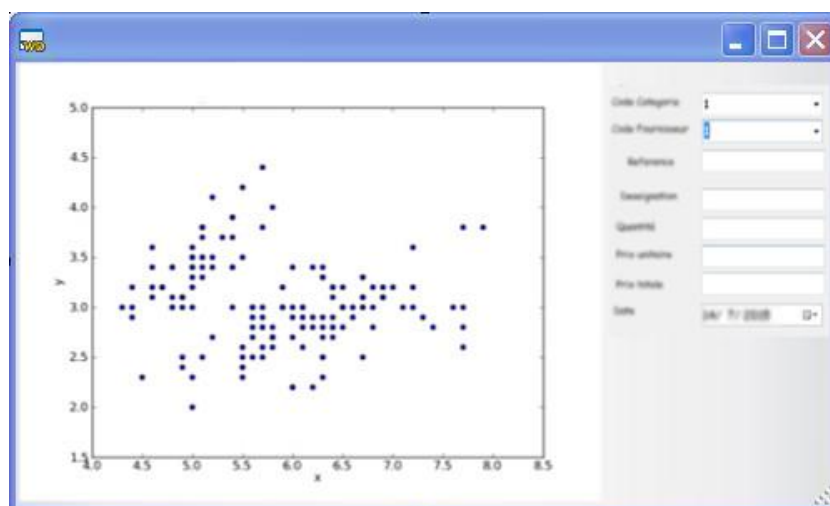


Illustration 42 : Maquette de l'interface homme/machine

### 1.3. Interfaces logiciel/logiciel

Les informations traitées par l'application sont contenues dans une base de données Oracle. L'application doit donc pouvoir communiquer avec cette base de données afin d'en extraire les informations nécessaires.

Dans un premier temps, cette communication se fera via des fichiers texte de type CSV. Ces fichiers texte contiendront les données extraites de la base de données. Les données seront ensuite extraites des fichiers par l'application puis seront utilisées pour l'analyse.

Dans un second temps, l'application sera directement reliée à la base de données Oracle et donc communiquera via des requêtes SQL.

## 2. Spécifications fonctionnelles

Dans cette partie est exposé les différentes fonctionnalités de l'application. Chaque fonctionnalité est détaillée de façon à en comprendre l'intérêt.

### 2.1. Définition de la fonction : Calculer une représentation graphique des données

Le rôle de cette fonction est de calculer un rendu pour l'affichage graphique des données que l'utilisateur souhaite consulter. Selon la représentation graphique choisie par l'utilisateur, elle se charge de déterminer la meilleure représentation graphique puis de la calculer.

Cette fonctionnalité est primordiale pour que l'application soit utilisable.

Elle a les spécificités d'entrées/sorties suivantes :

- Entrées : Liste des individus à afficher représenté sur N dimensions, choix de la représentation graphique
- Sortie : Tableau des points individus sur 2 dimensions

### 2.2. Définition de la fonction : Filtrer les données par caractéristiques des individus

Le rôle de cette fonction est de filtrer les individus afin d'alléger la représentation graphique. Son rôle est aussi de pouvoir mieux cerner la dissipation spatiale des individus par rapport à un critère qu'ils possèdent (par exemple, leur sexe).

Le filtrage se fera donc sur les critères définissant les individus en fonction de ce que l'on souhaite représenter ou non.

Cette fonctionnalité est moyennement prioritaire.

Elle a les spécificités d'entrées/sorties suivantes :

- Entrées : Liste des individus à filtrer représenté sur N dimensions correspondant aux critères des individus, les dimensions que l'on souhaite garder à la fin
- Sortie : Liste des individus filtrés

### 2.3. Définition de la fonction : Filtrer les données par indemnités

Le rôle de cette fonction est de filtrer les individus afin d'alléger la représentation graphique. Son rôle est aussi de pouvoir mieux cerner la dissipation spatiale d'une indemnité par rapport à une autre par exemple.

Le filtrage se fera donc sur les indemnités que l'on souhaite représenter ou non.  
Cette fonctionnalité est moyennement prioritaire.

Elle a les spécificités d'entrées/sorties suivantes :

- Entrées : Liste des individus à filtrer représenté sur N dimensions représentant les indemnités, les dimensions que l'on souhaite garder à la fin
- Sortie : Liste des individus filtrés

## **2.4. Définition de la fonction : Points communs entre individus**

Le rôle de cette fonction est de déterminer les points communs entre plusieurs individus. L'objectif principal de cette fonctionnalité est de connaître les facteurs influençant le calcul de chaque indemnité.

Cette fonctionnalité est d'une faible priorité.

Elle a les spécificités d'entrées/sorties suivantes :

- Entrée : Liste des individus sur N dimensions représentant leurs caractéristiques
- Sortie : Liste des facteurs communs

## **2.5. Définition de la fonction : Prédiction des individus potentiellement problématiques**

Le rôle de cette fonction est de prédire les individus qui peuvent être source de problème ou concerné par des erreurs de calculs sur la prochaine version du système LOUVOIS.

Cette fonctionnalité est d'une faible priorité.

Elle a les spécificités d'entrées/sorties suivantes :

- Entrée : Liste des individus
- Sortie : Individus pouvant être problématiques

## **2.6. Définition de la fonction : Sauvegarde des résultats**

Le rôle de cette fonctionnalité est de pouvoir sauvegarder les résultats obtenu sous forme d'un fichier PDF. Cela permet de conserver une trace des résultats obtenu sans à avoir à relancer tous les calculs.

Cette fonctionnalité est d'une priorité moyenne.

Elle a les spécificités d'entrées/sorties suivantes :

- Entrée : Données des analyses, éléments que l'utilisateur souhaite sauvegarder
- Sortie : Fichier PDF avec les analyses, graphiques, ...

## 3. Spécifications non fonctionnelles

Cette partie détaille les spécifications non fonctionnelles de l'application.

### 3.1. Contraintes de développement et conception

Afin de tester les différentes méthodes d'analyse de données ainsi que de tester le bon fonctionnement ou non des développements, un jeu de données est fourni par la société Sopra Steria.

Pour le moment, il n'y a pas de contrainte de langage de programmation ni d'environnement d'exécution de l'application. Cependant, il est nécessaire que l'environnement d'exécution dispose de la technologie CUDA d'NVIDIA pour que l'application fonctionne correctement.

### 3.2. Contraintes de fonctionnement et d'exploitation

#### 3.2.1. Performances

L'application doit présenter un haut niveau de performances compte tenu du nombre de données à analyser. En effet, pour que l'utilisateur puisse utiliser correctement l'application, celle-ci doit répondre rapidement à ses requêtes.

Une requête peut être un simple zoom dans une représentation graphique dans quel cas il faut que l'application soit très réactive (temps de réponse inférieur à 1 ou 2 secondes). Une requête peut aussi être une demande de changement de représentation graphique, de filtrage, d'agrégation, ... Dans ces cas, un temps de réponse plus long est toléré (inférieur à 10-20 secondes) puisque les opérations demandent davantage de calculs.

#### 3.2.2. Capacités

L'ordinateur utilisé pour analyser et afficher les résultats à l'utilisateur devra être suffisamment puissant pour que l'utilisateur ait un temps de réponse du logiciel le plus court possible.

Ses caractéristiques minimales ne sont pas encore totalement déterminées mais il devra posséder au minimum les éléments suivant :

- Processeur multi-cœurs (CPU) embarquant minimum 6 cœurs avec hyper threading
- De la mémoire vive (RAM), minimum 16Go
- Un processeur graphique (GPU) NVIDIA, embarquant la technologie CUDA
- Une mémoire de masse de type disque dur (HDD) ou de préférence de type solid-state drive (SSD)

#### 3.2.3. Contrôlabilité

Durant la phase de calcul, l'application ne doit pas être totalement bloquée. L'utilisateur doit pouvoir visualiser l'avancé du calcul et pouvoir l'arrêter s'il décide que celui-ci devient trop long.



---

### 3.2.4. Sécurité

L'application est accessible uniquement sur le poste sur lequel elle est installée ce qui signifie qu'elle n'est pas accessible depuis l'extérieur. De ce fait, il n'est pas prévu de phase d'authentification pour qu'un utilisateur puisse utiliser l'application.

## 4. Comptes rendus hebdomadaires (Weekly)

### Compte rendu n°1 du 20/09/2018

M. VENTURINI m'a présenté le projet avec davantage de détails que ce qui était présent dans l'énoncé du sujet. Le projet consiste à comparer les résultats de la version de production du système de paie LOUVOIS avec les résultats de la nouvelle version en développement. Le but de cette comparaison est de détecter d'éventuelles anomalies dans la version en développement du logiciel.

Afin de m'orienter vers une première piste, M. VENTURINI m'a présenté les résultats d'une de ses recherches qui consistait à trouver un moyen d'afficher rapidement une visualisation de plusieurs millions de lignes de données.

A l'issue de l'entretien, je me suis mis à la recherche d'informations concernant le projet en commençant par le logiciel LOUVOIS afin de bien comprendre son domaine d'action et les enjeux qu'il représente. J'ai ensuite approfondi la méthode de visualisation des données présentée par M. VENTURINI en étudiant son article. J'ai ensuite cherché d'autres moyens de visualiser de grandes quantités de données.

J'attends l'entretien avec M. Mickaël WINANDY mercredi prochain afin de connaître plus en détails le projet, les contraintes, les données disponibles, ainsi que l'objectif précis attendu. Cet entretien devra me permettre de rédiger une première version du cahier des charges.

### Compte rendu n°2 du 28/09/2018

Le mercredi 26 septembre s'est tenue une réunion de présentation du projet. Vous pouvez trouver le compte rendu de cette réunion en pièce jointe.

Suite à cette réunion, j'ai réalisé des recherches afin de me documenter sur l'état de l'art concernant la visualisation et le traitement de grandes quantités de données, notamment ceux évoqués lors de la réunion comme le data mining, le co-clustering, ...

La semaine prochaine je continuerai mes recherches. Aussi, j'ai rendez-vous avec M. Mikaël WINANDY jeudi à 14h00 à la caserne afin qu'il me fasse une présentation sur site.

### Compte rendu n°3 du 07/10/2018

Cette semaine, j'ai continué mes recherches sur l'état de l'art. J'ai aussi commencé à rédiger le cahier des charges.

Jeudi après-midi, j'ai eu un rendez-vous à la caserne de Tours avec M. Mikaël WINANDY qui m'a présenté le bâtiment où se situent les différents pôles dédiés aux développements et à la maintenance du logiciel LOUVOIS. Il m'a aussi présenté l'architecture de la base de données.

La semaine prochaine, continuerai l'état de l'art, ainsi que le cahier des charges.

**Compte rendu n°4 du 11/10/2018**

Cette semaine, j'ai fini la rédaction du cahier des charges (ci-joint). N'hésitez pas à me signaler les éléments manquant/incorrects.

Aussi, j'ai parcouru l'architecture de la base de données que M. WINANDY m'a envoyé afin de voir plus en détails comment elle était architecturée.

**Compte rendu n°5 du 19/10/2018**

Cette semaine, j'ai continué l'état de l'art en cherchant des méthodes de représentation graphique des données. Je me suis aussi renseigné sur quelques techniques de clusterisations des données.

La réunion de jeudi matin avec M. WINANDY m'a permis de faire un point sur mon actuel avancé ainsi que de définir des axes d'exploration plus spécifique.

La semaine prochaine, je commencerai à rédiger une première version du cahier des spécifications afin de pouvoir commencer à valider certains choix exploratoire.

**Compte rendu n°6 du 03/11/2018**

La semaine dernière ainsi que cette semaine, j'ai rédigé la première version du cahier des spécifications. Vous la trouverez joint à ce mail.

La semaine prochaine, nous avons réunion le mercredi 07 novembre à 15h30 à Polytech. M. WINANDY sera absent mais cette réunion sera l'occasion d'échanger sur le projet ainsi que sur le cahier des spécifications.

**Compte rendu n°7 du 08/11/2018**

La réunion de mercredi après-midi a été l'occasion d'échanger sur l'avancée du projet. Cela a aussi été l'occasion pour M. WINANDY de me transmettre une extraction anonymisée. M. VENTURINI a proposé de partir dans un premier temps sur une visualisation des individus. Pour réaliser cette visualisation il faut transformer les données en une matrice individus/indemnité pour ensuite faire du clustering sur cette matrice.

Suite à cette réunion je me suis mis à développer un petit outil en JAVA permettant de faire la transformation des données en matrice souhaitée. Cet outil est déjà fonctionnel bien qu'il faut encore optimiser certains points. Avec cette méthode les données d'entrées (fichier de 2.5Go) sont agrégées en moins de 40s (avec un processeur 12 cœurs et 4Go de RAM) en un fichier de 120 Mo.

La semaine prochaine, j'optimiserai quelques points de l'outil précédent afin de le rendre plus souple aux données et je commencerai à développer une méthode pour avoir assez rapidement une première visualisation des données.

**Compte rendu n°8 du 27/11/2018**

Il y a deux semaines, je n'ai pas pu travailler sur le PRD pour cause d'un entretien sur Paris le mercredi et du forum de Polytech le jeudi.

La semaine dernière, j'ai commencé la rédaction du rapport pour la soutenance du 12 ou du 13 décembre. J'ai aussi commencé à prendre en compte les remarques de Monsieur WINANDY sur le cahier des spécifications.

Cette semaine, je compte continuer la modification du cahier des spécifications afin de vous en transmettre une nouvelle version au plus tôt.

**Compte rendu n°9 du 29/11/2018**

Cette semaine, comme annoncé dans mon weekly précédent, j'ai modifié le cahier des spécifications en fonction des commentaires de M. WINANDY. Vous trouverez la nouvelle version en pièce jointe. J'ai identifié en rouge les éléments que j'ai modifiés ou ajoutés.

N'hésitez pas à me faire un retour sur ce qui va, ou ne va pas.

Pour la suite, je vais continuer à rédiger le rapport pour la soutenance du 12 ou du 13 décembre.

M. WINANDY, je profite de ce mail pour vous rappeler que vous pouvez assister à ma soutenance qui se tiendra le matin du 12 ou du 13 décembre. Je n'ai pas eu de réponse de votre part si vous souhaitiez y assister, ceci dans le but de réserver le créneau de passage qui vous arrange le mieux.

**Compte rendu n°10 du 17/12/2018**

La semaine dernière s'est déroulée ma soutenance. Le mercredi a donc été consacré à la préparation de cette soutenance. Le jeudi, après la soutenance, a eu lieu une réunion pour redéfinir la direction du projet, notamment les premiers éléments à implémenter.

La réunion a permis d'établir que nous allons nous concentrer sur les représentations des évolutions des indemnités dans le temps. L'idée est, dans un premier temps, de reproduire l'existant. On y ajoutera ensuite des options de filtrage sur les indemnités ou la mise en place d'un seuil minimal pour qu'une indemnité soit affichée.

Puisqu'une équipe de Sopra Steria sera susceptible de reprendre le projet, celui-ci sera développé en Python.

Mercredi 19 prochain, une nouvelle réunion avec M. WINANDY se déroulera à partir de 15h afin de développer d'avantage cette méthode de visualisation.

J'ai aussi rendez-vous avec M. VENTURINI pour tester des méthodes de représentation des individus.

**Compte rendu n°11 du 17/01/2019**

Durant la semaine dernière, j'ai commencé à chercher des librairies de visualisation pour Python et pour Java. La librairie Matplotlib de Python semble la plus complète pour le projet. Je n'ai pas effectué davantage de travail puisque j'ai consacré une partie du temps à finir d'autres projets

pédagogiques qui avaient une date de rendu proche.

Cette semaine, j'ai rencontré M. Mikaël WINANDY lors du cours de planification et suivi de projet. Nous avons pu discuter de ma gestion du projet. Aussi, il m'a demandé de découper le travail en petits livrables. Vous trouverez donc en pièce jointe la description du premier livrable.

La semaine prochaine, j'ai rendez-vous mercredi à 10h00 avec M. Mikaël WINANDY pour que l'on puisse échanger sur ce premier livrable.

#### **Compte rendu n°12 du 25/01/2019**

Durant la réunion avec Monsieur WINANDY, nous avons évoqués les différentes possibilités de représentations et de filtrages des données que l'on pouvait réaliser. J'ai recopié la totalité dans le document PDF joint. Ces possibilités ne seront pas toutes intéressantes à explorer. Il faudra déterminer nos priorités dans la réunion de la semaine prochaine.

Aussi, en attendant d'avoir les données, j'ai commencé à faire quelques fonctions pour transformer les données en une structure acceptée par la librairie Matplotlib.

M. WINANDY, serait-il possible d'apporter les données que je vous ai demandées la semaine dernière ?

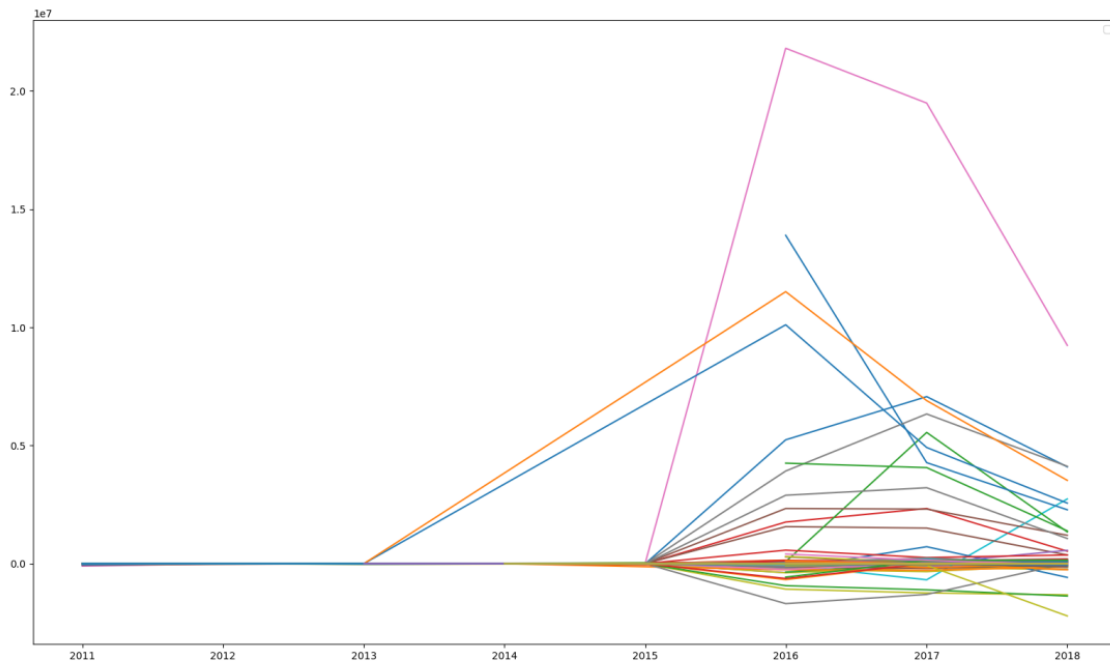
La semaine prochaine, le rendez-vous de mercredi 9h00 va permettre d'échanger à propos du premier livrable et des priorités de représentations et de filtrages à réaliser. Puis je continuerais à développer l'application.

#### **Compte rendu n°13 du 04/02/2019**

Durant la semaine, j'ai continué à développer une première version de l'application avec matplotlib ainsi qu'avec des données pour pouvoir faire des tests. Vous trouverez en pièce jointe le résultat de cette visualisation. Cette visualisation représente l'évolution des indemnités au cours des années. On peut remarquer qu'il y a un certain nombre d'indemnités qui n'ont pas de données avant 2016. On peut aussi remarquer que les données que l'on a avant 2016 ne sont pas toutes complètes et sont toutes proche de la valeur 0.

La semaine du 04 février, j'ai une réunion sur le site de l'armée pour avoir une nouvelle présentation du fichier Excel qui est utilisé actuellement pour l'analyse de l'évolution des indemnités. Cela me permettra de me remémorer les différents éléments qui sont utilisés pour l'analyse.

**Pièce jointe :**



**Compte rendu n°14 du 10/02/2019**

Mercredi matin, j'ai eu une réunion avec Monsieur Brun Benjamin qui m'a fait une présentation de la procédure de tests. Il m'a aussi refait une présentation sur l'utilisation du fichier Excel. Cette réunion m'a été très utile pour mieux comprendre vers quoi je dois aller. Elle a aussi été l'occasion de me réexpliquer certains points dont je n'en avais pas saisi pleinement le principe lors de ma première visite.

Dans la suite de la semaine, j'ai continué à réaliser l'application pour le premier livrable. Je n'ai pas beaucoup avancé car j'ai passé beaucoup de temps à comprendre et à prendre en main une librairie permettant de faire des fenêtres graphique facilement.

La semaine prochaine me permettra de continuer l'application. J'espère avancer rapidement maintenant que je commence à maîtriser l'outil graphique. Je pense avoir le temps d'implémenter une fonctionnalité de filtrage sur la date par exemple.

**Compte rendu n°15 du 04/03/2019**

Ce compte rendu synthétise le travail que j'ai réalisé durant les trois dernières semaines (semaines 07, 08, et 09).

Comme annoncé dans le weekly précédant, j'ai réalisé une première version de l'application pour la semaine 09 (du 25 février).

Comme présentée le mercredi 27 février à M. WINANDY et à M. RAGOT, l'application permet de visualiser l'évolution dans le temps des indemnités. L'application permet aussi d'appliquer différents filtres comme :

- Filtrer par version de LOUVOIS
- Filtrer par date (intervalle de temps)
- Filtrer par le nombre d'indemnités, ou bien des indemnités en particulier en fonction de leur nom

L'application permet de visualiser les données d'une version en particulier (capture d'écran 1) ou bien de visualiser une comparaison de deux versions (capture d'écran 2).

Durant la présentation plusieurs points ont été soulevés comme par exemple sélectionner les indemnités visionnées par leurs plus grandes différences plutôt que par leurs valeurs brutes. Aussi, l'application doit posséder une fonctionnalité de « drill down » pour aller explorer les données plus en détail (par exemple, consulter tous les individus qui représentent une indemnité sur un intervalle de temps donné).

Dans la suite de la réunion nous avons décidé de développer la fonctionnalité de « drill down ». Pour cela, M. WINANDY va me fournir des données (valeur de chaque administré, mois par mois, pour les 10 indemnités qui ont une différence la plus importante LOUVOIS). De mon côté, je dois explorer une manière de représenter les individus graphiquement, ainsi que de mettre en évidence les axes discriminant des données. L'utilisation d'une méthode de type AFCM est envisagée (équivalent à une ACP mais pour des colonnes symboliques)

Après la réunion, j'ai consacré ma semaine à implémenter l'affichage des données dans un tableau (capture d'écran 3) afin de pouvoir les explorer. Cela permet d'avoir un complément à la visualisation.

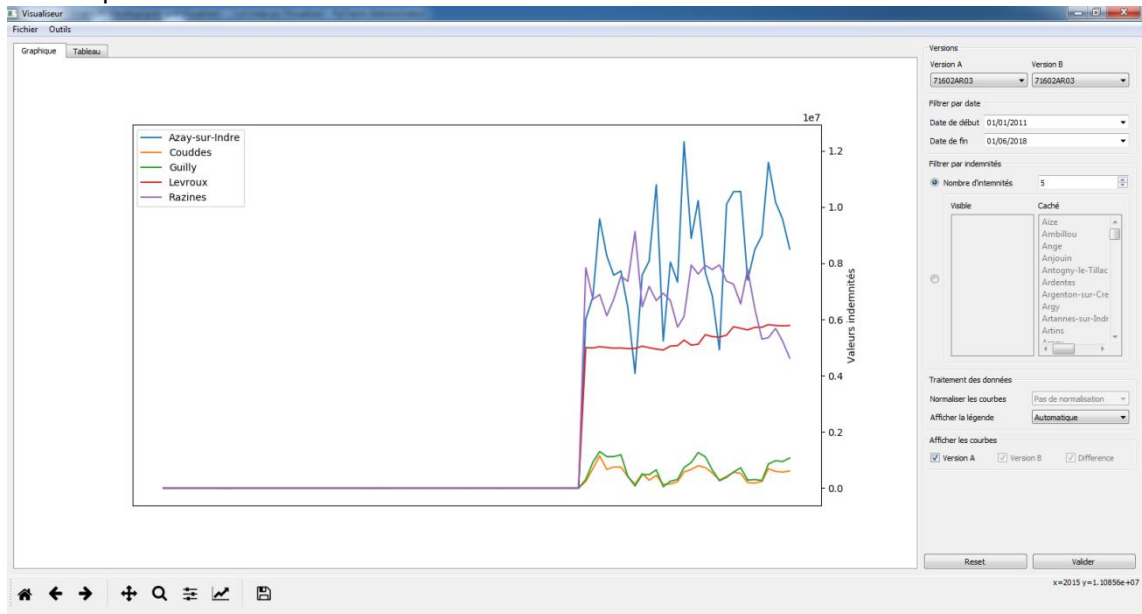
Je suis ensuite passé à la réalisation de la phase de « drill down », et j'ai commencé par représenter les données dans un tableau. Face à la grande quantité de données, je me suis confronté à un problème de surutilisation de la mémoire RAM. J'ai dû trouver des contournements afin d'en optimiser son utilisation.

La semaine 10 (4 mars), une réunion est prévue pour faire l'échange des données. Ce sera aussi l'occasion d'échanger sur ce que j'ai réalisé depuis le rendez-vous précédant.

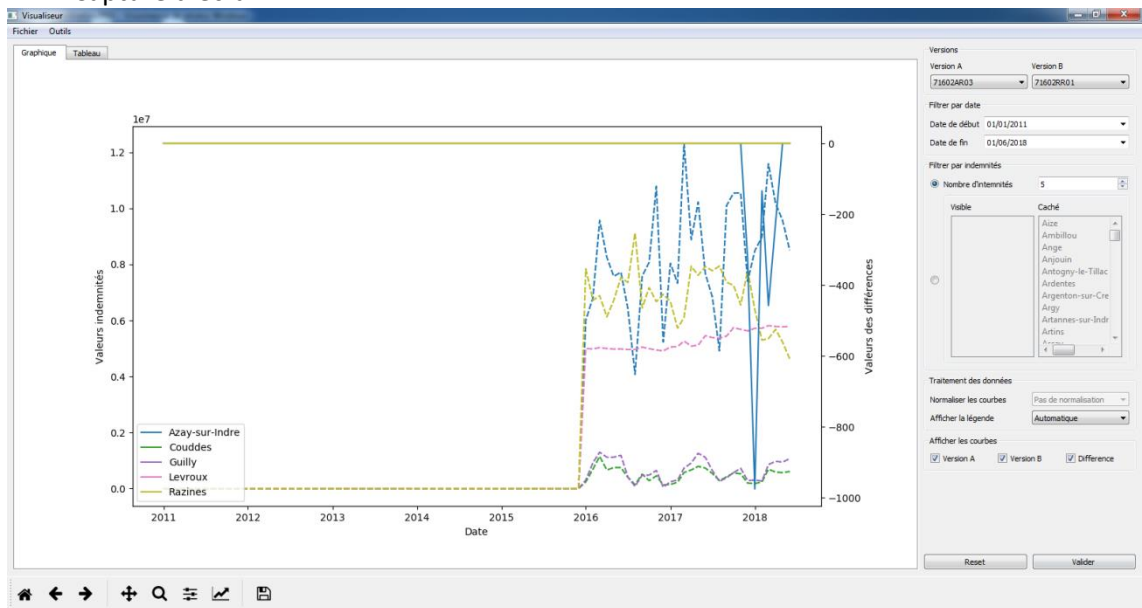
Cette semaine sera l'occasion de continuer à développer l'application en y implémentant une analyse AFCM par exemple.

## Pièces jointes :

### - Capture d'écran 1



### - Capture d'écran 2





### - Capture d'écran 3

The screenshot shows a software application window titled 'Visualiseur'. It contains a table with columns: Date, Indemnité, Valeur version A, Valeur version B, Différence, and Différence absolue. The table lists 22 rows of data for various locations like Housay, Mers-sur-Indre, and Cerelles. To the right of the table is a sidebar with filters for 'Versions' (Version A and B), 'Filtrer par date' (Date de début and Date de fin), 'Filtrer par indemnités' (Nombre d'indemnités), and 'Traitement des données' (Normaliser les courbes, Afficher la légende, and Afficher les courbes). The sidebar also has a list of locations under 'Visible' and 'Caché'.

	Date	Indemnité	Valeur version A	Valeur version B	Différence	Différence absolue
1	2018/01/01	Housay	-41425647.00	-41429262.00	42615.00	42615.00
2	2018/03/01	Housay	-416875537.00	-416922180.00	46643.00	46643.00
3	2018/04/01	Housay	-417211952.00	-417255213.00	43261.00	43261.00
4	2017/12/01	Housay	-415295862.00	-415337375.00	41413.00	41413.00
5	2018/01/01	Mers-sur-Indre	327175686.40	327213456.10	-37769.70	37769.70
6	2018/03/01	Mers-sur-Indre	3261487115.50	326283711.40	-36997.90	36997.90
7	2018/05/01	Housay	-416465838.00	-416500511.00	34673.00	34673.00
8	2018/04/01	Mers-sur-Indre	329397776.80	329411802.30	-34315.50	34315.50
9	2017/12/01	Mers-sur-Indre	328012109.20	328036049.20	-32840.00	32840.00
10	2018/06/01	Housay	-41668536.00	-416717813.00	30787.00	30787.00
11	2018/05/01	Mers-sur-Indre	328889342.40	328916845.10	-27502.70	27502.70
12	2018/02/01	Housay	-415030477.00	-415057955.00	27478.00	27478.00
13	2018/06/01	Mers-sur-Indre	329664205.40	329688626.00	-24420.60	24420.60
14	2018/02/01	Mers-sur-Indre	327756453.60	327748250.40	-21796.50	21796.50
15	2017/11/01	Housay	-415125837.00	-415139889.00	14062.00	14062.00
16	2018/03/01	Cerelles	176913.07	190619.68	-13706.61	13706.61
17	2018/06/01	Binas	-28200835.50	-28212270.80	11435.10	11435.10
18	2017/11/01	Mers-sur-Indre	327849441.20	327861104.20	-11153.60	11153.60
19	2018/05/01	Binas	-28065267.30	-28075689.60	10422.30	10422.30
20	2017/11/01	Cerelles	190715.95	199853.69	-9137.74	9137.74
21	2018/05/01	Cerelles	185012.49	192322.66	-7310.17	7310.17
22	2018/03/01	Binas	-28181812.40	-28189115.80	7303.40	7303.40

### Compte rendu n°16 du 12/03/2019

Comme annoncé dans mon précédent weekly, j'ai continué à développer l'application pour intégrer une analyse de données. Durant la réunion de mercredi dernier, nous avons discuté de réaliser une analyse des données via une analyse en composantes principales (ACP). Cette analyse permet de d'obtenir la représentation spatiale des individus en fonction de leur valeur pour chaque indemnité ainsi que de voir la corrélation entre chaque indemnité. Aussi, avoir une option permettant d'inverser la visualisation individus <=> indemnités a été jugée pertinent.

Actuellement, j'ai réalisé une partie de l'implémentation de l'ACP (résultats vérifiés avec le logiciel R). Un début de visualisation a été réalisé pour la répartition des individus (capture d'écran 1) ainsi que la corrélation entre les variables (capture d'écran 2). Cependant, il me reste la fonctionnalité d'inversion de la visualisation, ainsi que diverses fonctions de protection pour que le calcul de l'ACP s'effectue dans les meilleures conditions.

Cette semaine, je vais donc réaliser cela afin d'avoir une application pleinement exploitable, notamment sur la partie ACP.

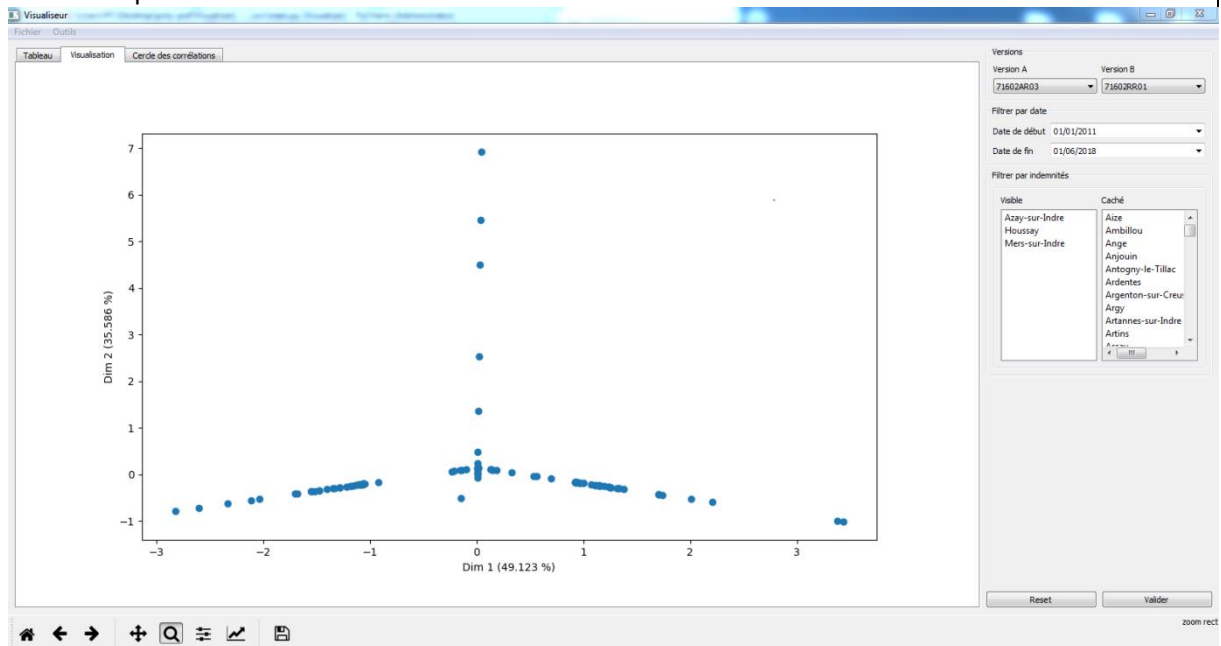
Ce weekly est aussi l'occasion pour moi de vous communiquer la date de ma soutenance. Elle se déroulera le jeudi 28 mars 2019 entre 9h30 et 10h15. M. WINANDY, merci de me signaler votre présence.

Pour finir, une réunion est prévue avec M. WINANDY le mercredi 20 mars à 9h00 pour faire le point sur ce que j'ai développé. Ce sera la dernière réunion du projet avant la soutenance. M. RAGOT n'étant pas disponible le mercredi 20, un entretien est prévu le mardi 19 dans la matinée.

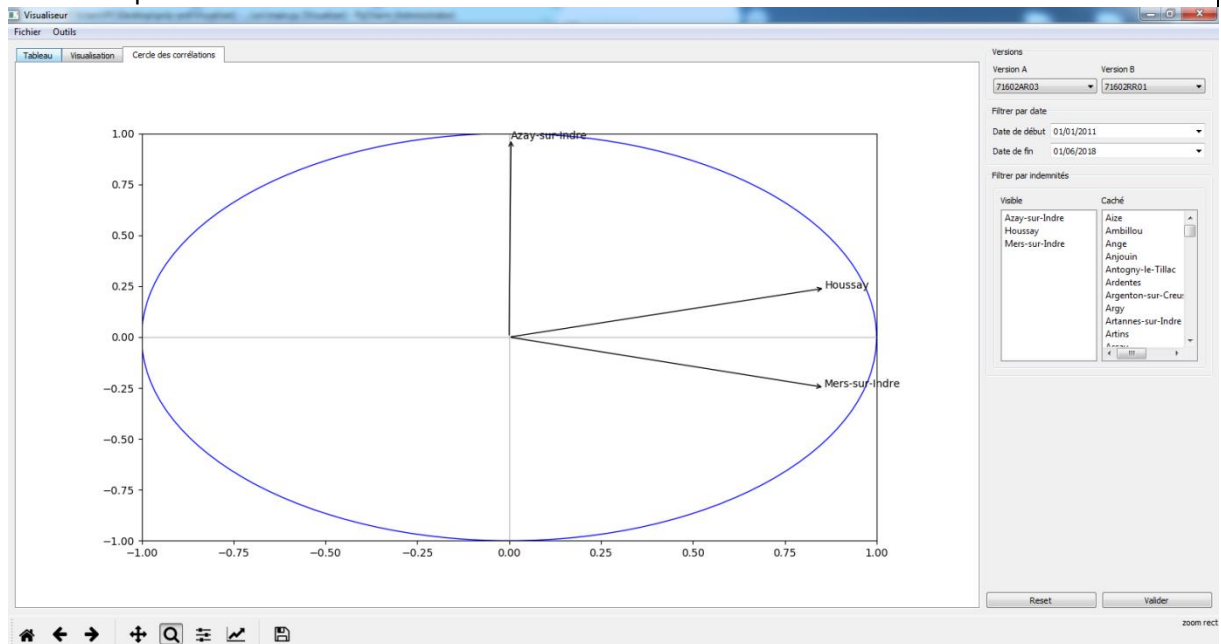
M. VENTURINI, vous pouvez vous joindre aux réunions selon vos disponibilités.

**Pièces jointes :**

- Capture d'écran 1



- Capture d'écran 2



**Compte rendu n°17 du 17/03/2019**

Cette semaine j'ai continué mon implémentation de l'ACP dans l'application. Aussi j'ai correctement lié le passage des données entre la visualisation globale et la visualisation de drill down.

Pour compléter mon précédent weekly ainsi que répondre au mail de M. RAGOT, je réalise l'analyse ACP sur les individus filtrés en fonction de ce qu'a renseigné l'utilisateur dans la barre de droite de l'application. Ensuite, l'ACP s'effectue sur la valeur de la différence entre les deux versions.

Chacun des individus, ainsi que chacune des indemnités ont le même poids. Les indemnités sont normalisées afin qu'il n'y ait pas de phénomène d'écrasement.

Dans l'exemple de mon weekly précédent, on a effectivement une représentation de 85% de l'inertie ce qui peut paraître peu pour seulement 3 axes mais cela vient des données. Avec d'autres données il est possible d'avoir de meilleures représentations. Ensuite, concernant les corrélations, on peut dire que Houssay et Mers-sur-Indre sont relativement liés et donc l'origine de l'écart pourrait avoir la même source, tandis d'Azay-sur-Indre doit avoir un facteur indépendant des deux autres indemnités.

On pourra en parler d'avantage durant notre réunion de la semaine prochaine.

M. RAGOT, j'en profite pour vous demander le moment ou vous souhaiteriez faire la réunion mardi matin (il n'y pas d'heure qui a été fixé).

---

## 5. Document utilisateur

### 5.1. Introduction

Ce document présente comment installer, et utiliser l'application.

Les étapes d'installation sont fonctionnelles sous le système d'exploitation Windows 7 x64. Cependant, l'application doit pouvoir fonctionner sur d'autres versions de Windows, ainsi que sur d'autres systèmes d'exploitation, comme Linux par exemple.

Ce document a pour but d'expliquer l'utilisation de l'application. Il n'explique cependant pas comment interpréter les résultats obtenus.

## 5.2. Installation

### 5.2.1. Python

L'application a été développée avec le langage Python. Pour fonctionner, elle nécessite donc un interpréteur Python. Il est possible de télécharger Python sur le site officiel : [www.python.org](http://www.python.org).

L'application a été développée avec la version 3.7.2 de Python. Il faut donc choisir de préférence cette version, ou plus récent. Il est cependant possible que l'application fonctionne avec des versions antérieures de Python.

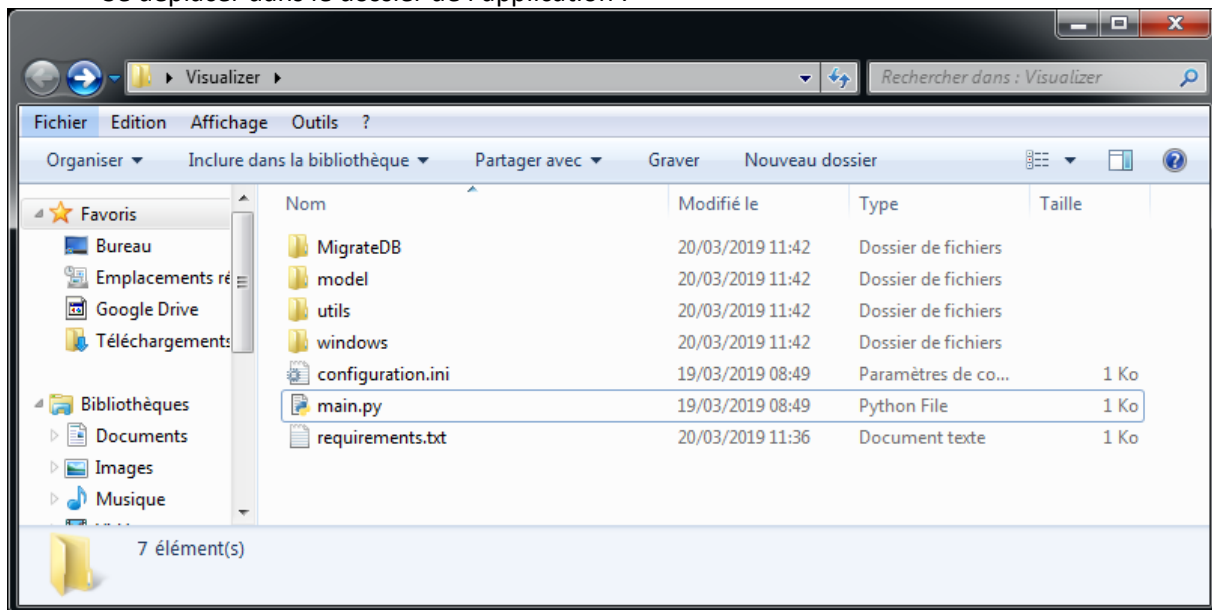
Il suffit d'installer Python en suivant l'installateur par défaut, les configurations proposées par défaut sont suffisantes.

### 5.2.2. Dépendances de l'application

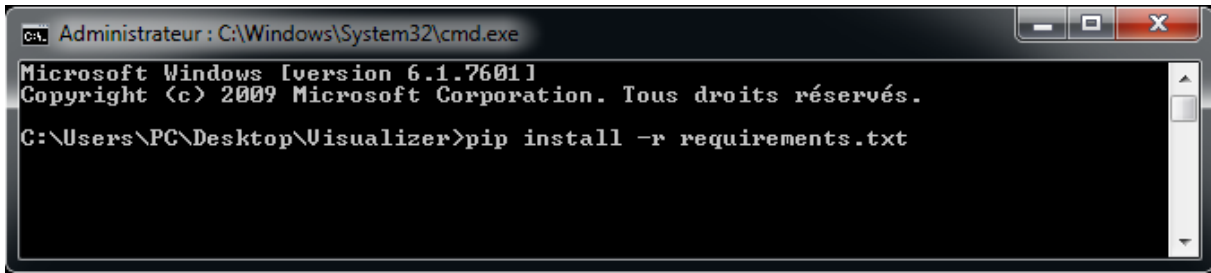
L'application utilise des libraires externes pour fonctionner. Il faut donc installer ces libraires pour que l'application puisse s'en servir.

Pour installer ces dépendances, il faut exécuter la commande : « `pip install -r requirements.txt` ». Pour cela, il faut suivre les étapes suivantes :

- Se déplacer dans le dossier de l'application :



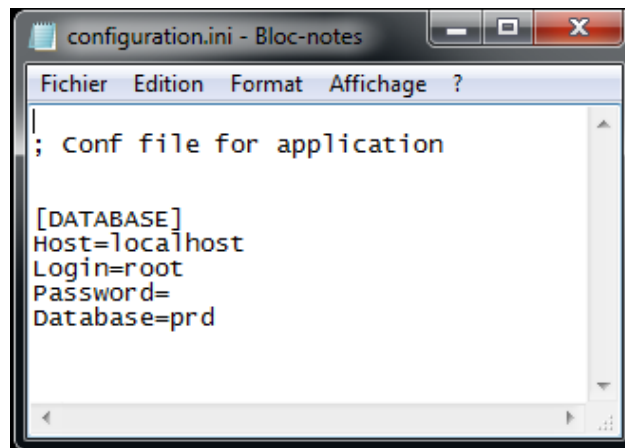
- Il faut ensuite ouvrir la console. Pour réaliser cela il existe 2 façons :
  - o Taper « cmd » dans la barre d'adresse de l'explorateur de fichier
  - o Faire shift + click droit dans une zone vide du dossier, puis, dans le menu contextuel, cliquer sur « Ouvrir une fenêtre de commandes ici »
- Dans la fenêtre de console, taper la commande « `pip install -r requirements.txt` ».



```
Administrateur : C:\Windows\System32\cmd.exe
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.
C:\Users\PC\Desktop\Visualizer>pip install -r requirements.txt
```

### 5.2.3. Configuration

L'application possède un fichier de configuration permettant de facilement changer les paramètres. Ce fichier de configuration se situe à la racine du dossier de l'application et se nomme : « configuration.ini ».

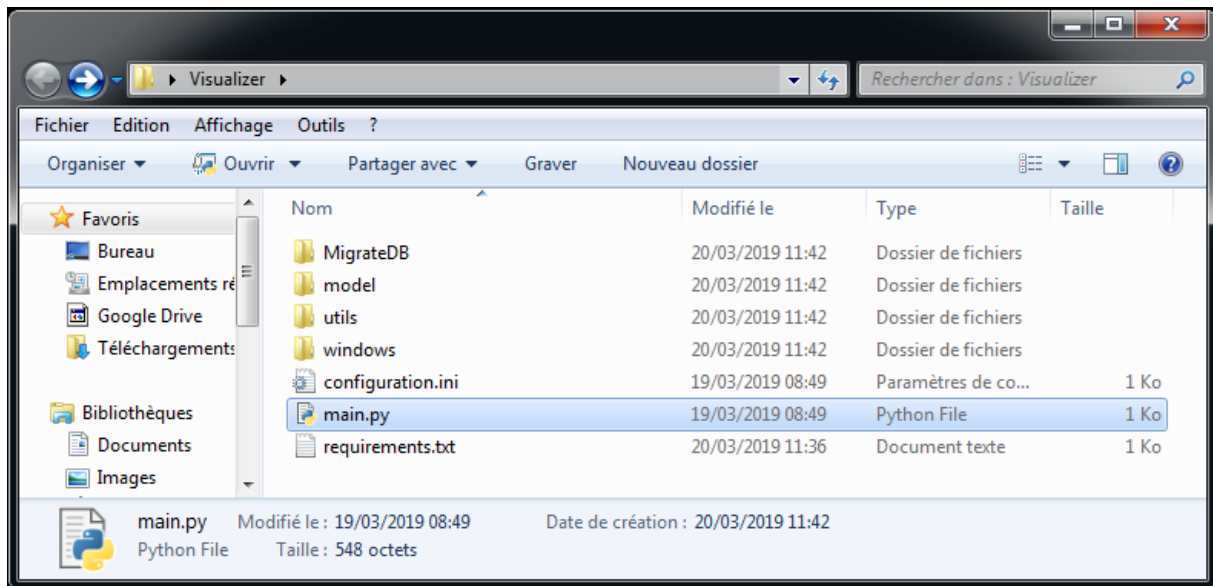


Pour fonctionner, l'application a besoin de se connecter à une base de données MySQL. Le fichier de configuration permet de régler l'adresse IP de connexion (Host), le nom d'utilisateur (Login), le mot de passe (Password), ainsi que le nom de la base de données (Database).

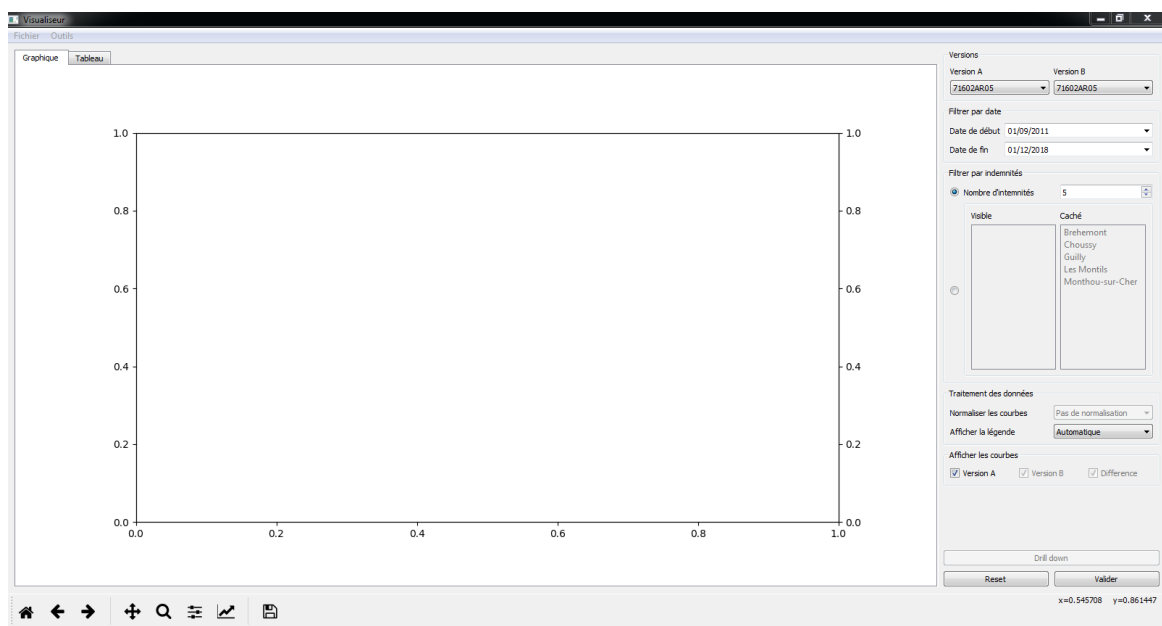
## 5.3. Utilisation

### 5.3.1. Lancer l'application

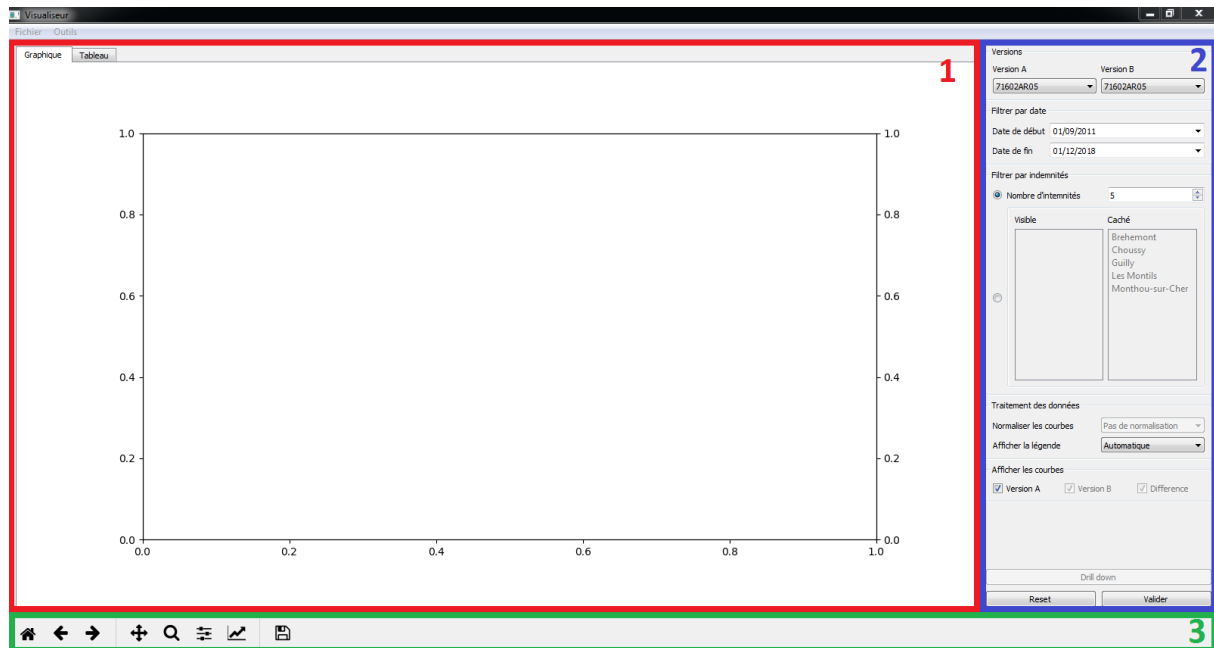
Pour lancer l'application, il suffit d'exécuter le fichier « main.py » se situant à la racine du dossier. Pour l'exécuter, il est possible de le faire via la console en tapant le nom du fichier (cela permet d'avoir la sortie d'erreur de l'application), ou plus simplement de double cliquer sur le fichier.



Si l'installation a correctement été effectuée, alors l'application va démarrer en affichant la fenêtre principale qui est la suivante :



### 5.3.2. Décomposition de la fenêtre



La fenêtre principale se décompose en trois grandes zones :

- La **zone 1** est la zone de visualisation des données. On y retrouve deux onglets. Le premier onglet est pour l'affichage de courbes, et le second onglet contient un tableau avec toutes les données qui sont utilisées pour l'affichage du premier onglet.
- La **zone 2** est la zone permettant de filtrer les résultats. Cette zone se décompose en plusieurs autres zones qui seront détaillées plus tard dans ce document. Dans cette zone, on retrouve notamment la possibilité de filtrer les données en fonction de leurs dates, des indemnités, de la version des résultats, ....
- La **zone 3** est la zone permettant de naviguer dans le graphique présent dans la **zone 1**. Cela permet de zoomer, se déplacer, ou encore exporter la visualisation en cours.

### 5.3.3. Utilisation de la fenêtre principale

Lorsque l'application vient de se lancer, la fenêtre n'affiche rien dans la zone du graphique. Pour afficher des données, il faut utiliser la zone de filtrage pour sélectionner les bons paramètres, puis appuyer sur valider pour que l'application prenne en compte les filtres et affiche les données dans le graphique et dans le tableau.

Les valeurs présentent dans l'outil de filtrages sont des valeurs qui sont extraites à partir de la base de données. En cas d'ajout/suppression de données dans la base données, il est possible de



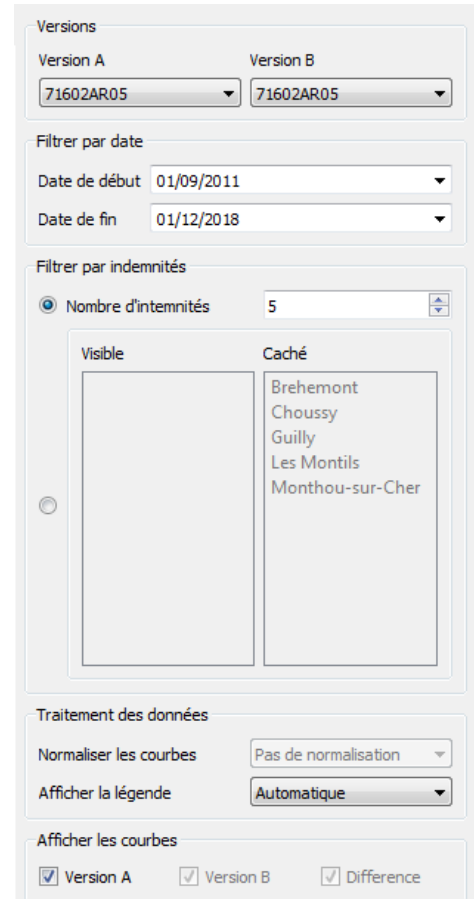
demander à l'application de recalculer ces valeurs en allant dans Outils → Réinitialiser le cache.

Pour restaurer les valeurs initiales de l'outil de filtrage, il suffit d'appuyer sur le bouton reset.

Un bouton « drill down » permet d'explorer les individus qui sont utilisés pour réaliser la représentation. Pour faire cette étude, une seconde fenêtre est ouverte.

Les options de filtrages sont les suivantes :

- Versions : Il est possible de choisir la version que l'on souhaite afficher. Dans ce cas, il faut que la version A corresponde à la version B.  
Lorsque la version A est différente de la version B, alors l'application affichera les données pour la version A et B ainsi que les différences entre les deux versions.
- Date : Grace au filtrage par date, il est possible de ne visualiser les données que sur une fenêtre de temps.
- Indemnités : Il est possible de choisir entre l'affichage d'un certain nombre d'indemnités, ou une sélection d'indemnités.
  - Filtrage par le nombre d'indemnités : Dans ce filtrage, il est possible de préciser le nombre d'indemnités que l'on souhaite visualiser. L'application va alors afficher les indemnités qui ont la plus forte valeur dans le cas d'une visualisation avec une seule version, ou alors les indemnités qui ont les plus fortes différences dans le cas d'une comparaison entre deux versions.
  - Filtrage par sélection d'indemnités : Ce filtre permet de choisir spécifiquement des indemnités. La liste « caché » contient toutes les indemnités présentes en base de données. Un double clic sur une indemnité permet de la faire basculer dans la liste « visible ». Seule la liste visible est prise en compte pour l'affichage.
- Traitement des données : Cela permet de normaliser ou non les courbes, ainsi que d'afficher ou de masquer la légende. (option non fonctionnelle)
- L'affichage des courbes permet de masquer des versions spécifiques. Si la case à cocher est sélectionnée, alors les données en rapport avec la version seront affichées.



The screenshot shows a web-based interface for data filtering and processing. It includes sections for selecting versions (A and B), filtering by date (start and end dates), filtering by the number of indemnities (set to 5), and a list of indemnities categorized as 'Visible' or 'Caché'. The 'Caché' list includes Brehemont, Choussy, Guilly, Les Montils, and Monthou-sur-Cher. There are also options for data treatment (normalizing curves, showing legend) and checkboxes for displaying specific versions (A, B, and Difference).

Lorsque l'on appuie sur le bouton valider, l'application charge les données depuis la base de données puis les affiche dans la fenêtre de visualisation. Les données chargées sont aussi insérées dans l'onglet du tableau.

## Affichage des différences deux versions :



Si l'on souhaite masquer certaines indemnités sans recharger toutes les données, il est possible de cliquer sur la ligne de l'indemnité dans la légende.

## Affichage du tableau des données :

	Date	Indemnité	Valeur version A	Valeur version B	Différence	Différence absolue
1	2017/03/01	Les Montils	3333570.60	3852153.17	-518582.57	518582.57
2	2017/01/01	Les Montils	3314107.32	3826269.72	-512162.40	512162.40
3	2017/02/01	Les Montils	3324774.83	3835633.54	-510858.71	510858.71
4	2017/04/01	Les Montils	3316019.61	3816655.09	-500635.48	500635.48
5	2017/05/01	Les Montils	3316387.60	3815531.82	-499144.22	499144.22
6	2017/06/01	Les Montils	3320799.28	3811968.69	-491169.41	491169.41
7	2017/07/01	Les Montils	3282530.40	3756993.50	-474463.10	474463.10
8	2017/05/01	Guilly	1251531.21	1626528.08	-374996.87	374996.87
9	2018/07/01	Brehemont	4782311.81	5110748.82	-328437.01	328437.01
10	2017/07/01	Brehemont	4834089.51	4563838.84	270250.67	270250.67
11	2017/11/01	Les Montils	359227.13	3772276.62	-213049.49	213049.49
12	2018/04/01	Les Montils	3596675.06	3808971.56	-212296.50	212296.50
13	2018/03/01	Les Montils	3614956.92	3827227.32	-212270.40	212270.40
14	2018/01/01	Les Montils	3598887.02	3810973.76	-212086.74	212086.74
15	2018/05/01	Les Montils	3585802.70	3796318.00	-210515.30	210515.30
16	2018/02/01	Les Montils	358960.31	3799527.18	-209566.87	209566.87
17	2017/08/01	Les Montils	3485574.38	3693324.70	-207750.32	207750.32
18	2018/07/01	Les Montils	3516823.82	3723298.91	-206475.09	206475.09
19	2017/12/01	Les Montils	3594469.86	3800083.32	-205613.46	205613.46
20	2017/10/01	Les Montils	3502006.28	3706867.81	-204861.53	204861.53
21	2017/09/01	Les Montils	3480968.09	3685765.48	-204797.39	204797.39
22	2017/06/01	Guilly	1110925.52	1315662.33	-204736.81	204736.81

Certaines cases sont colorées. La coloration correspond à la grandeur de la valeur de la case par rapport au minimum et au maximum dans la colonne. Les valeurs négatives sont affichées en orange, tandis que les valeurs positives sont affichées en rouge. Plus la valeur de la case est proche du minima ou du maxima, plus la couleur sera intense.

Un clic sur l'entête des colonnes permet de trier la colonne par ordre croissant puis décroissant.

### 5.3.4. Utilisation de la fenêtre de drill down

La fenêtre de drill down permet de réaliser une analyse ACP sur les individus. Elle est accessible uniquement lorsqu'on sélectionne deux versions différentes car elle utilise les différences entre les deux versions pour réaliser l'analyse. Cette fenêtre est indépendante de la fenêtre principale ce qui signifie que l'on peut fermer la fenêtre principale sans que cela ferme cette fenêtre.

Cette fenêtre se décompose en trois zones, comme pour l'application principale. Cependant, différentes visualisations, ainsi que d'autres filtres sont disponibles.

Dans la zone de filtrage ce qui change principalement est la possibilité de choisir de faire l'ACP sur les individus ou sur les indemnités.

Cette fenêtre dispose aussi d'un tableau de sortie pour observer les valeurs des individus. Étant donné qu'il y a une grande quantité d'individus, l'affichage des données dans le tableau peut demander beaucoup de ressources et beaucoup de temps. Il est donc possible de désactiver cette sortie.

ACP

ACP sur :

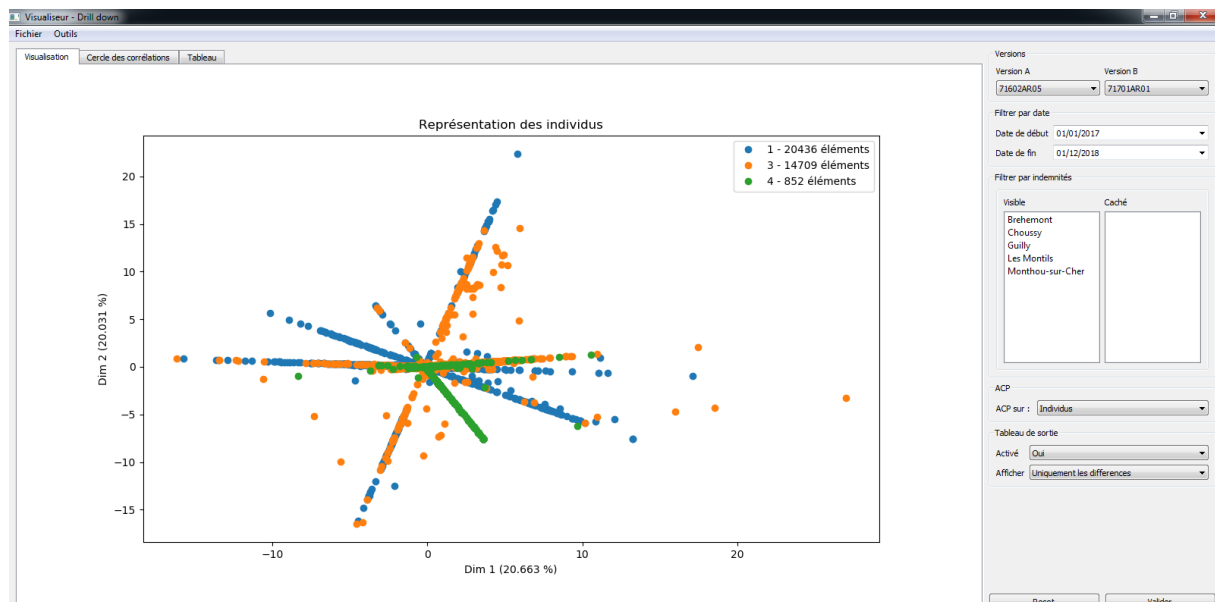
Tableau de sortie

Activé

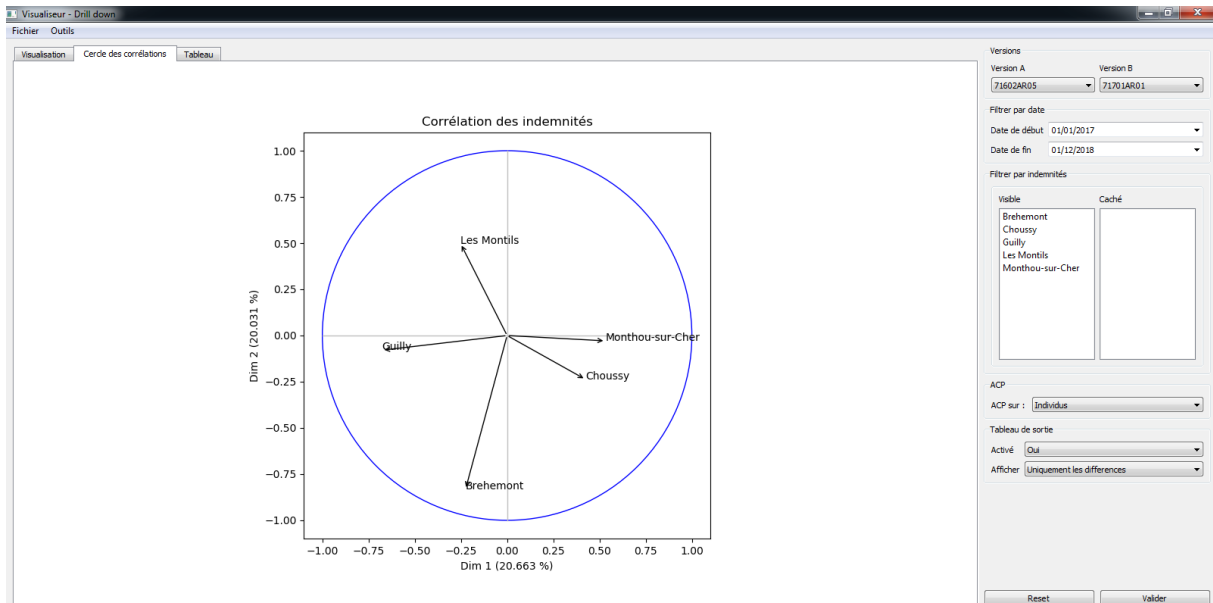
Afficher

La visualisation principale permet d'observer la répartition des individus dans l'espace. Les individus sont colorés en fonction de leur corps d'armée d'appartenance (SIRH). Le nombre d'individu représenté dans chaque corps d'armée est disponible dans la légende.

La légende des axes contient le pourcentage de représentation des données fourni par l'axe. Dans notre exemple, l'axe 1 fournit 20,663% des informations initiales des données.



Un deuxième onglet permet d'observer la corrélation entre les indemnités. Le graphique de cet onglet ne permet pas de navigation dans celui-ci (zoom, déplacement, ...). Cette visualisation permet aussi de connaître l'importance des axes dans la représentation.



Le troisième onglet, permet d'afficher la valeur des individus. Comme pour la fenêtre principale, les cases sont colorées en fonction de leur valeur.

Le tableau a un maximum de 100 000 lignes pour éviter un temps de chargement trop long et une trop grosse consommation de RAM.

The screenshot shows the 'Tableau' tab of the 'Visualiseur - Drill down' application. The table displays individual data with the following columns: Id administré, SRH, Indemnité, Date, Valeur version A, Valeur version B, Différence, and Différence absolue. The table is color-coded based on the 'Différence' column. The right sidebar contains the same filters as the previous screenshot.

	Id administré	SRH	Indemnité	Date	Valeur version A	Valeur version B	Différence	Différence absolue
1	216044	1	Brehemont	2018-08-01	-4488.65	14314.81	-18803.46	18803.46
2	214297	1	Brehemont	2017-10-01	17474.19	0.00	17474.19	17474.19
3	323890	3	Brehemont	2017-07-01	17150.85	0.00	17150.85	17150.85
4	82271	1	Brehemont	2017-05-01	-410.63	16159.48	-16570.12	16570.12
5	70250	1	Brehemont	2017-05-01	-410.63	16159.48	-16570.12	16570.12
6	221522	1	Brehemont	2017-07-01	16026.21	0.00	16026.21	16026.21
7	26285	1	Brehemont	2017-05-01	-869.57	14782.67	-15652.24	15652.24
8	239430	1	Brehemont	2017-07-01	15051.51	0.00	15051.51	15051.51
9	324886	3	Brehemont	2017-07-01	15051.51	0.00	15051.51	15051.51
10	234920	1	Brehemont	2018-07-01	14751.61	0.00	14751.61	14751.61
11	232057	1	Brehemont	2017-06-01	14676.63	0.00	14676.63	14676.63
12	206112	1	Brehemont	2017-07-01	14676.63	0.00	14676.63	14676.63
13	223670	1	Brehemont	2017-06-01	14245.52	0.00	14245.52	14245.52
14	236129	1	Brehemont	2018-07-01	14245.52	0.00	14245.52	14245.52
15	1097	1	Brehemont	2018-08-01	13027.15	0.00	13027.15	13027.15
16	11803	1	Brehemont	2017-07-01	-3172.06	9153.16	-12325.22	12325.22
17	282310	3	Brehemont	2017-07-01	-2973.62	8970.57	-11944.19	11944.19
18	232499	1	Brehemont	2017-07-01	11597.91	0.00	11597.91	11597.91
19	203772	1	Brehemont	2018-07-01	11597.91	0.00	11597.91	11597.91
20	262439	1	Brehemont	2018-07-01	-2819.07	8647.25	-11466.32	11466.32
21	279908	3	Brehemont	2018-07-01	-2882.42	8457.20	-11339.62	11339.62
22	335304	3	Brehemont	2017-07-01	10684.14	0.00	10684.14	10684.14

---

## 6. Document développeur

### 6.1. Introduction

Ce document présente l'ensemble des actions nécessaires afin d'obtenir un environnement de développement permettant de faire fonctionner le projet.

Cette documentation a été rédigée lors d'un développement dans un environnement Windows 7 64 bits. Il est donc possible que quelques étapes diffèrent selon le système d'exploitation et l'environnement de développement.

Ce document a donc pour but d'expliquer la configuration nécessaire au développement, ainsi que de présenter globalement l'architecture du projet.

## 6.2. Installation

### 6.2.1. Python

L'application a été développée avec le langage Python. Pour fonctionner, elle nécessite donc un interpréteur Python. Il est possible de télécharger Python sur le site officiel : [www.python.org](http://www.python.org).

L'application a été développée avec la version 3.7.2 de Python. Il faut donc choisir de préférence cette version, ou plus récent. Il est cependant possible que l'application fonctionne avec des versions antérieures de Python.

Il suffit d'installer Python en suivant l'installateur par défaut, les configurations proposées par défaut sont suffisantes.

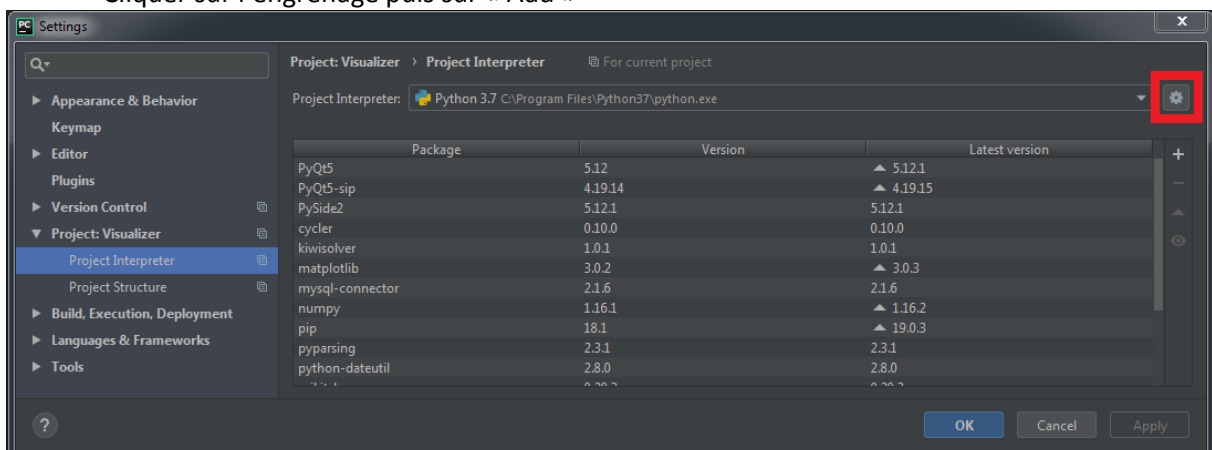
### 6.2.2. Environnement de développement (IDE)

Le projet a été réalisé à l'aide de l'IDE [PyCharm](https://www.jetbrains.com/pycharm/) de l'éditeur Jet Brains. Il est tout à fait possible d'utiliser un autre IDE bien que je recommande PyCharm.

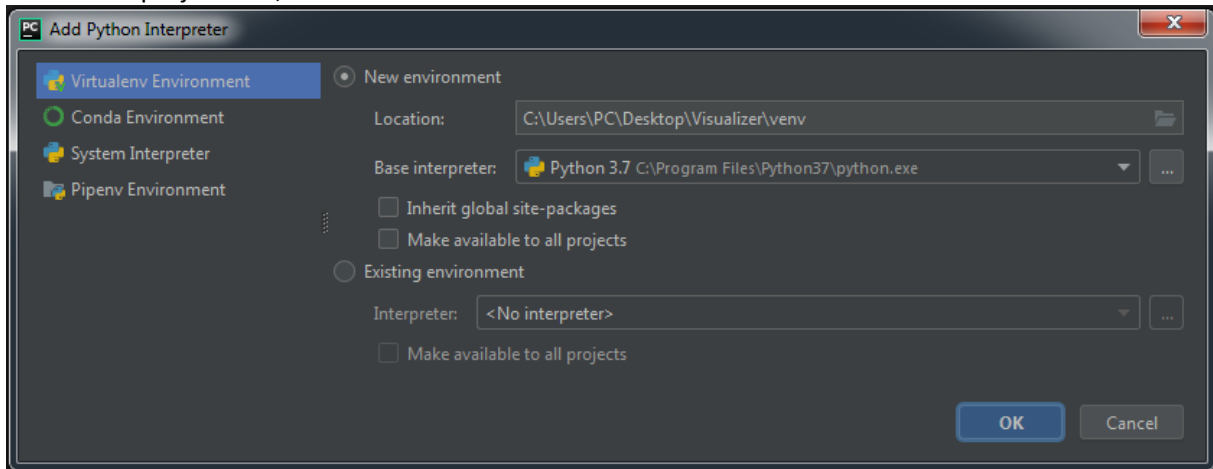
- Importer le projet en spécifiant le dossier du projet.
- Avec un clic droit sur le dossier src, faire « Mark Directory as » → « Sources Root ».

PyCharm permet de mettre en place un environnement virtuel. Cet environnement virtuel est très pratique car il permet d'avoir une isolation complète du projet par rapport aux autres projets. Je recommande donc de le mettre en place. Pour ce faire :

- File → Settings → Project → Project Interpreter
- Cliquer sur l'engrenage puis sur « Add »



- Choisir « New environment » et définir son chemin avec le dossier « venv » dans le dossier du projet. Puis, valider

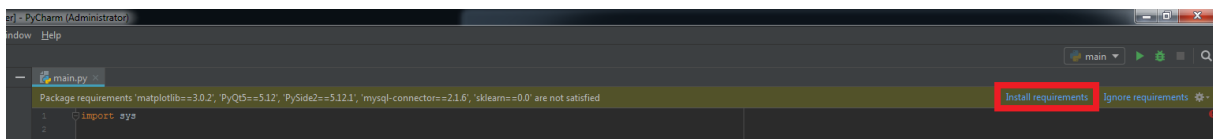


- Avec un environnement virtuel de configuré, il est nécessaire d'utiliser les outils proposés par PyCharm pour qu'ils fonctionnent. Par exemple, les commandes doivent être effectuées dans le terminal PyCharm et pas dans un terminal externe car sinon elles ne fonctionnent pas. Cela s'explique par le fait que PyCharm utilise l'environnement virtuel, ce que ne fait pas un simple terminal de commandes.

### 6.2.3. Dépendances de l'application

L'application utilise des libraires externes pour fonctionner. Il faut donc installer ces librairies pour que l'application puisse s'en servir.

PyCharm gère les dépendances et est capable de dire lorsqu'une dépendance est manquante. Il se base sur la liste des dépendances dans le fichier « requirements.txt » qui est à la racine du projet. Si PyCharm est correctement configuré, lorsqu'on se rend dans un fichier du code du projet, comme le fichier main.py, PyCharm propose de télécharger les dépendances manquantes :



Si PyCharm ne propose pas de lui-même d'installer les dépendances, il est possible de le faire via le terminal intégré a PyCharm via la ligne de commande : « `pip install -r requirements.txt` ».

#### 6.2.4. Base de données

Une base de données est nécessaire à l'application. En effet, l'application tire ses données depuis la base de données.

La base de données doit être de type MySQL. Elle contient 2 tables : *indemnities\_by\_month* et *users\_by\_month*.

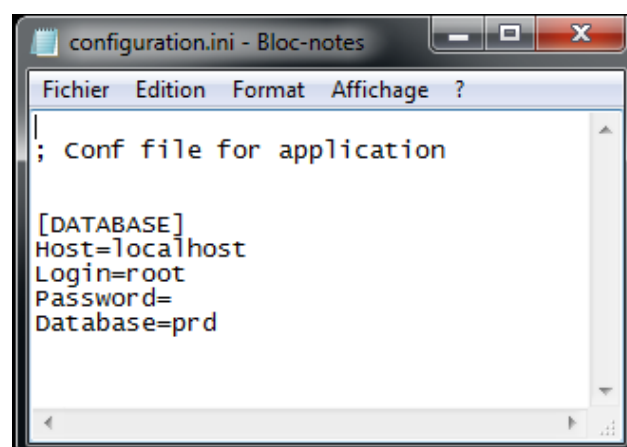
La table *indemnities\_by\_month* permet de stocker la valeur de chaque indemnité, mois par mois. C'est une agrégation des utilisateurs pour un mois et une indemnité donnée. Cette table est utilisée pour la fenêtre principale.

La table *users\_by\_month* permet de stocker la valeur des indemnités pour un mois et un administré donnée. Cette table est normalement très volumineuse (plusieurs Giga-octets) et est utilisé par la fenêtre de drill down.

indemnities_by_month	users_by_month
id:int(11) [PRIMARY, AI] version:varchar(255) sirh:int(11) calc_type:varchar(255) caba_type:varchar(255) value_type:int(11) date:date value:double	id:int(11) [PRIMARY, AI] version:varchar(255) sirh:int(11) idUser:int(11) calc_type:varchar(255) caba_type:varchar(255) value_type:int(11) date:date value:double

#### 6.2.5. Configuration

L'application possède un fichier de configuration permettant de facilement changer les paramètres. Ce fichier de configuration se situe à la racine du dossier de l'application et se nomme : « configuration.ini ».



Pour fonctionner, l'application a besoin de se connecter à une base de données MySQL. Le fichier de configuration permet de régler l'adresse IP de connexion (Host), le nom d'utilisateur (Login), le mot de passe (Password), ainsi que le nom de la base de données (Database).



### 6.3. Architecture

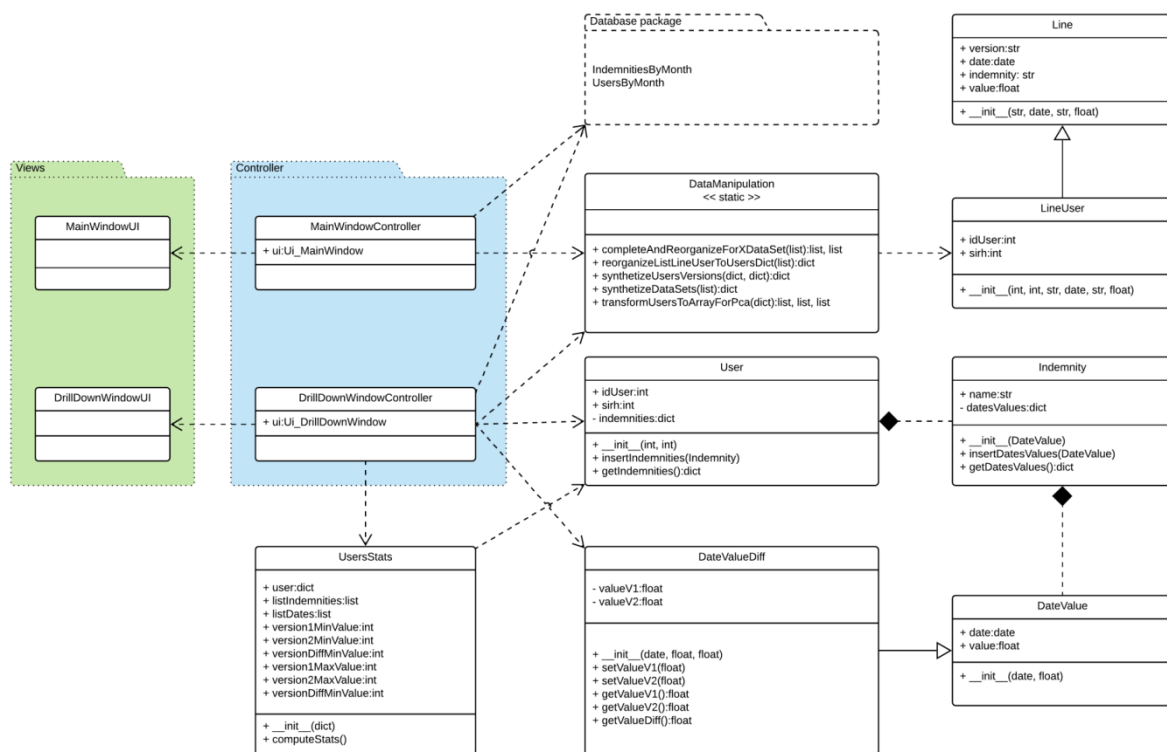
L'application suit le pattern MVC. C'est-à-dire, qu'il y a des classes gérant le stockage des données, des classes gérant l'affichage des fenêtres, et des classes permettant de faire la liaison entre les deux précédant types de classes.

Les vues et les contrôleurs sont rangés dans le dossier « windows ». Ce dossier contient les dossiers correspondant aux vues et aux contrôleurs. Il contient aussi une classe ayant le rôle de stoker des fonctions communes aux contrôleurs.

Le dossier « model » contient les classes permettant de gérer le stockage des données ainsi que les classes des objets de l'application.

Le dossier « utils » contient des fonctions utilitaires. Ces fonctions manipulent des objets et sont utilisées dans les contrôleurs pour réaliser des traitements sur les données.

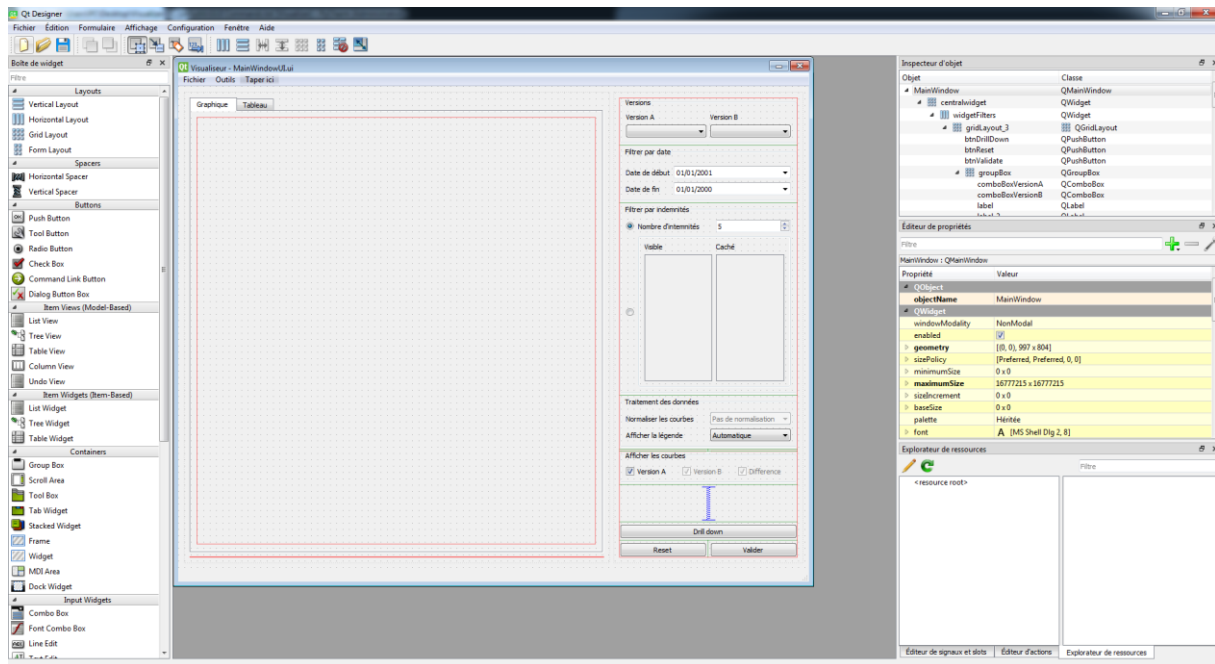
Le diagramme de classe de l'application est le suivant :



## 6.4. Modifier les vues

Les vues sont réalisées avec la librairie Qt. La librairie Qt embarque un utilitaire Qt Designer. Cet utilitaire permet de créer ses vues de façon graphique. Il exporte ensuite un fichier .ui.

Pour lancer Qt Designer, il faut taper la commande « `venv\Lib\site-packages\PySide2\designer.exe` » dans le terminal de PyCharm. S'ouvre alors l'utilitaire. En sélectionnant le fichier .ui de la fenêtre principale on obtient la capture d'écran ci-après.



Une fois les modifications effectuées, il faut exécuter un autre utilitaire pour convertir le fichier .ui en fichier Python .py.

Cet utilitaire s'exécute avec la commande :

« `venv\Scripts\pyside2-uic.exe [Chemin fichier .ui] [Chemin fichier .py] -x` ».

Le fichier généré ne doit jamais être modifié, car lors d'une nouvelle génération, les modifications seront perdues. Pour modifier/ajouter des comportements à la fenêtre, il faut le faire dans un autre fichier. C'est en grande partie le rôle des classes des contrôleurs.

## 6.5. Tests unitaires

L'application a été testée grâce à des tests unitaires. Les tests unitaires se trouvent dans le dossier test du projet.

Les tests qui ont été réalisés couvrent seulement les fonctions de manipulation des données. Le code des contrôleurs, des vues, ainsi que du model (qui gère le stockage des données) n'a pas été testé pour des raisons de manque de temps. Il est cependant possible de réaliser des tests. En effet Qt embarque un ensemble de fonctions permettant d'en réaliser sur les vues.

Les tests unitaires peuvent être lancé via PyCham (clic droit sur le dossier test → run 'Unittests in test') ou via un module Python. Ce module s'appelle « nose » et s'installe avec la commande : « *pip install nose* ». Pour lancer les tests il suffit d'exécuter la commande « *nosetests -v* ». Si tout se passe bien, le résultat est le suivant :

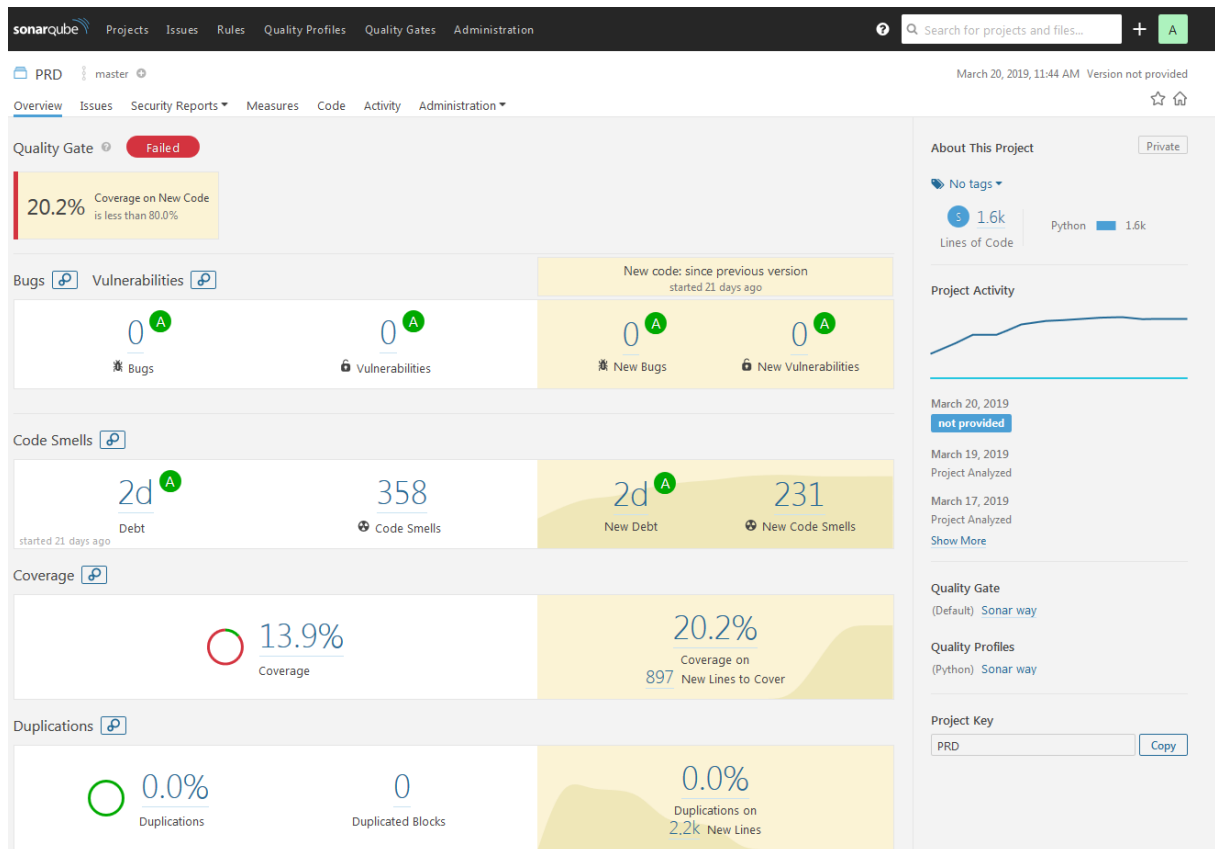
```
(venv) C:\Users\PC\Desktop\Visualizer>nosetests -v
test_dataValue (test_DateValue.test_DataManipulation) ... ok
test_dataValueDiff (test_DateValueDiff.test_DataManipulation) ... ok
test_dataValueDiffChangeV1 (test_DateValueDiff.test_DataManipulation) ... ok
test_dataValueDiffChangeV2 (test_DateValueDiff.test_DataManipulation) ... ok
test_Indemnity (test_Indemnity.test_Indemnity) ... ok
test_Line (test_Line.test_Line) ... ok
test_LineUser (test_LineUser.test_LineUser) ... ok
test_User (test_User.test_User) ... ok
test_computeStats (test_UsersStats.test_UsersStats) ... ok
test_completeAndReorganizeForXDataSet (test_Config.test_Config) ... ok
test_completeAndReorganizeForXDataSet (test_dataManipulation.test_DataManipulation) ... ok
test_reorganizeListLineUserToUsersDict (test_dataManipulation.test_DataManipulation) ... ok
test_synthesizeDataSets (test_dataManipulation.test_DataManipulation) ... ok
test_synthesizeUsersVersions (test_dataManipulation.test_DataManipulation) ... ok
test_transformUsersToArrayForPca (test_dataManipulation.test_DataManipulation) ... ok

-----
Ran 15 tests in 0.106s

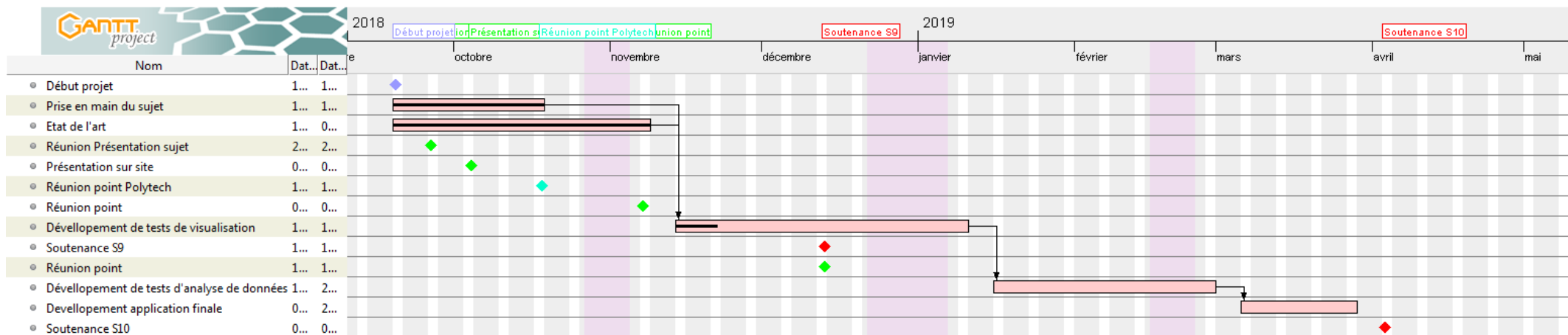
OK

(venv) C:\Users\PC\Desktop\Visualizer>
```

## 7. Rapport SonarQube



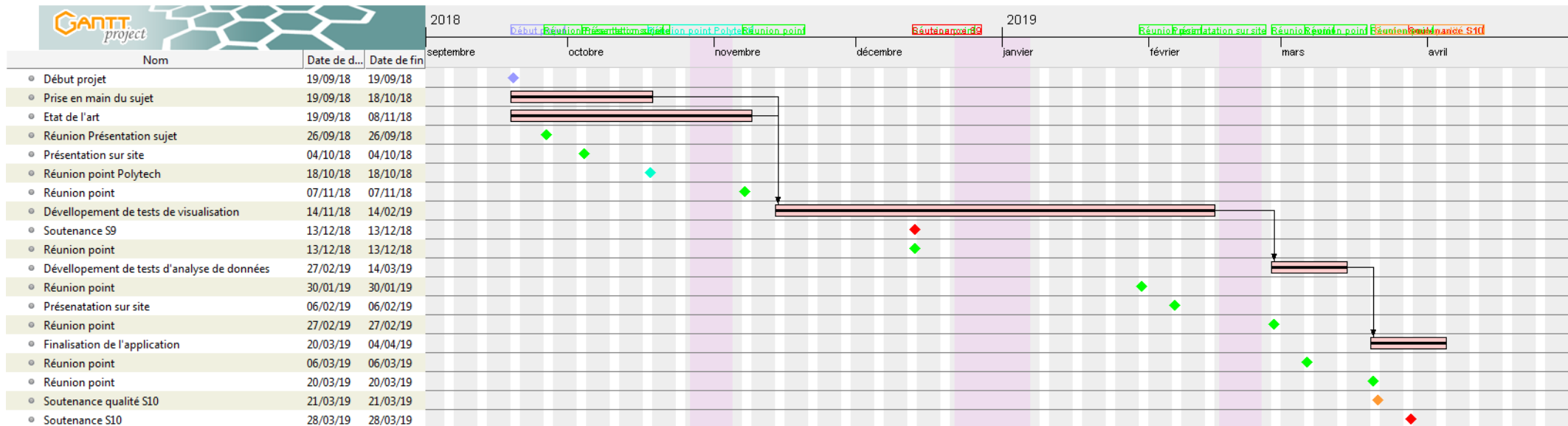
## 8. Diagramme de Gantt prévisionnel



Ce diagramme est le diagramme prévisionnel depuis le précédent jusqu'à la fin du projet.

Les jalons en vert symbolisent les rendez-vous avec les tuteurs académiques et le tuteur entreprise. Lorsque que le jalon est bleu, cela signifie que le rendez-vous s'est déroulé uniquement avec les tuteurs académiques. Les jalons en rouge représentent des dates clés du projet.

## 9. Diagramme de Gantt réel



Ce diagramme est le diagramme réel. On remarque qu'il y a une tâche qui diffère par rapport au planning prévisionnel : « développement de tests de visualisation ». En effet, cette tâche a été bien plus longue que prévu. Cela s'explique par le fait qu'il y a eu plus de représentation graphique que prévu et que la prise en main de la bibliothèque graphique a été plus compliquée que prévu.

# Détection préventive de bugs dans le versement de la solde de l'armée

**CONFIDENTIEL**

## Résumé

Ce projet de recherche et de développement est en partenariat avec l'entreprise Sopra Steria. Ce projet porte sur l'évolution du logiciel LOUVOIS. Le logiciel LOUVOIS est un logiciel développé par Sopra Steria et destiné à calculer la solde des militaires français. Ce logiciel gère près de 200 000 militaires et près de 300 indemnités ce qui en fait un logiciel très complexe. Le but de ce projet est de détecter des régressions (anomalies) lors des mises à jour du logiciel LOUVOIS.

Ce projet porte donc sur l'analyse des données de sortie du logiciel pour détecter des anomalies. Ce rapport permet de décrire le projet, les problématiques, ainsi que les recherches effectuées pour réaliser ce projet.

Mots-clés : LOUVOIS, analyse de données, visualisation de données, K-Mean, DBSCAN, clustering

## Abstract

This research and development project is in partnership with Sopra Steria. This project concerns the evolution of the LOUVOIS software. LOUVOIS software is a software developed by Sopra Steria to calculate the pay of French military members. This software manages about 200,000 military personnel and about 300 allowances, making it a very complex software. The purpose of this project is to detect regressions (anomalies) during updates of the LOUVOIS software.

This project focuses on analyzing the software's output data to detect anomalies. This report describes the project, the issues, as well as the research conducted to carry out this project.

Keywords : LOUVOIS, data mining, visual data mining, K-Mean, DBSCAN, clustering

**Entreprise**  
Sopra Steria

**Etudiant**  
Guillaume SERVAIS (DI5)

**Tuteur entreprise**  
Mikaël WINANDY

**Tuteurs académique**  
Nicolas RAGOT  
Gilles VENTURINI