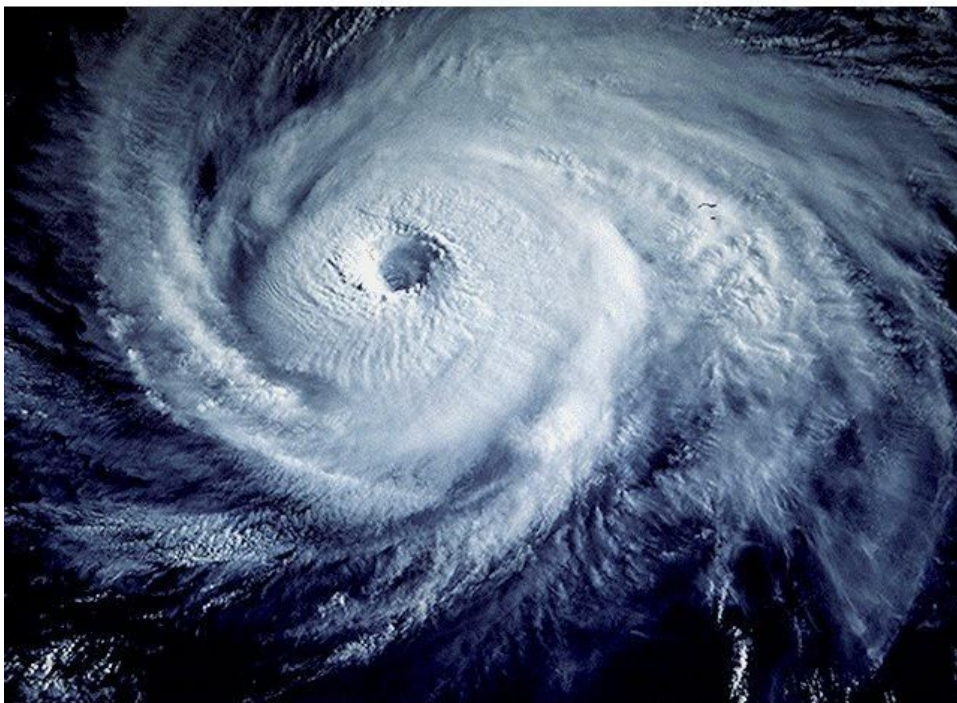


# Les catastrophes naturelles



Rédigé par :

Nicolas Brouté

Killian Buhr

Noémie Deloeuvre

Projet tableau de bord  
Année universitaire 2017 - 2018

## Versions du rapport

Date	Version	Parties modifiées	Cause
19 / 02 / 2018	V 1.0	Toutes	Début du rapport
26 / 02 / 2018	V 1.1	Parties 1, 2, 3, 4	Explication sujet
05 / 03 / 2018	V 1.2	Parties 5.1, 7	Finition charte de codage
12 / 03 / 2018	V 1.3	Parties 5.2, 6	Scripts et schéma de la BDD
19 / 03 / 2018	V 1.4	Parties 5.3, 8	Ajout des résultats obtenus
26 / 03 / 2018	V 1.5	Toutes	Dernière relecture et correction

## Sommaire

1. Objet du document
2. Problématique du sujet
3. Document de préférences et terminologie
4. Gestion de projet
5. Démarche de projet
  - 5.1. Recherche d'informations et préparation de données
    - 5.1.1. Recherche de données
    - 5.1.2. Collecte de données
    - 5.1.3. Nettoyage des données
  - 5.2. Administrer
    - 5.2.1. Stockage de données
    - 5.2.2. Interrogation de données
  - 5.3. Visualisation de données
    - 5.3.1. Méthodes statistiques
    - 5.3.2. Dashboard Excel
    - 5.3.3. Carte HTML
6. Gestion de configuration
  - 6.1. Script de création de la base de données
  - 6.2. Procédures
7. Assurance qualité et contrôle qualité
  - 7.1. Charte de codage
    - 7.1.1. Convention de nommage des fichiers
    - 7.1.2. Convention de nommage des variables et fonctions
    - 7.1.3. Commentaires du code
    - 7.1.4. Partie codage
  - 7.2. Revues
  - 7.3. Tests
8. Bilan
  - 8.1. Bilan Client
  - 8.2. Bilan Fournisseur

## Remerciements

Nous tenons tout d'abord à remercier Madame Wahiba Bashoun pour ses conseils avisés et les cours qu'elle nous a donné qui ont permis le bon déroulement de ce projet. Nous la remercions également pour l'encadrement qu'elle nous a fourni tout au long de la réalisation.

Nous remercions également Monsieur Riad Mokadem pour ses conseils sur la base de données, ses conseils sur la faisabilité de notre projet ainsi que pour son encadrement.

Nous tenons également à remercier l'ensemble de l'équipe pédagogique que nous avons eu durant ces deux années à l'université Paul Sabatier, qui nous ont permis d'obtenir les connaissances nécessaires à la réalisation de ce projet.

Enfin nous remercions toutes les personnes, qui de près ou de loin, ont contribué à la réussite de ce projet ainsi qu'à l'écriture du rapport.

## 1. Objet du document

L'objectif de cette partie est de présenter les différents contributeurs, les clients ainsi que le contexte du projet.

Nous sommes étudiants en Master 1 Statistiques et Informatique Décisionnelle. Dans le cadre de notre formation nous avons l'opportunité de mettre en application différentes techniques de conception, de programmation et de gestion de projet apprises au cours de notre cursus.

L'objectif de notre projet est de développer un système d'aide à la décision élaboré par des analyses détaillées sur des points d'intérêts à partir de la visualisation de données textuelles issues d'une base de données en ligne.

Les principaux clients des résultats obtenus est l'équipe encadrante du projet : Monsieur Riad Mokadem et Madame Wahiba Bashoun.

## 2. Problématique du sujet

Depuis des centaines d'années, la planète est en proie à de nombreux phénomènes météorologiques naturels (éruption volcanique, séisme, ouragan...). Ces phénomènes se sont amplifiés à cause des activités humaines qui ont accéléré le réchauffement climatique. Après leurs passages, les dégâts peuvent être importants et des victimes sont souvent à déplorer. Ces catastrophes font donc l'actualité dans les médias lorsqu'ils se produisent.

Notre problématique est la suivante : Les catastrophes naturelles sont-elles plus fréquentes et destructrices depuis le début du XXI<sup>e</sup> siècle?

Afin d'essayer de répondre à notre problématique, nous allons nous limiter à quelques catastrophes qui sont les inondations, les séismes, les ouragans, les typhons, les cyclones et les tremblements de terres. Nous allons exploiter des milliers d'articles de 5 sources différentes (Le Monde, Le Figaro, Le Point, Nouvel Obs, Libération) depuis l'année 2000.

### 3. Documents de références et terminologie

#### 3.1. Documents de références

Pour mener à bien ce projet nous avons utilisé le cours de Base de Données (BD) introduit par Monsieur Morvan en Licence 3. Il nous a servi à modéliser le schéma entité - association de notre base de données. Pour requêter celle-ci nous nous sommes servi du support du cours SQL SERVER présenté au début du projet.

Durant la démarche de projet nous avons été amenés à utiliser le score TF IDF (de l'anglais term frequency-inverse document frequency). Il nous a été présenté également en Licence 3 au cours d'Extraction d'Informations par Madame Lechani.

C'est une méthode de pondération utilisée dans la fouille de textes. Cette mesure statistique permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus. Le poids augmente proportionnellement au nombre d'occurrences du mot dans le document. Il varie également en fonction de la fréquence du mot dans le corpus.

En ce qui concerne la partie gestion de projet, assurance et contrôle qualité, nous avons utilisé nos connaissances acquises grâce aux cours de Génie logiciel.

#### 3.2. Terminologie

Afin de mieux comprendre notre problématique, voici les définitions des 6 catastrophes naturelles que nous avons décidé d'étudier.

Les cyclones, les typhons et les ouragans sont le même type de catastrophe naturelle mais sont situés à des endroits différents du globe. Ils désignent une grande zone où l'air atmosphérique est en rotation autour d'un centre de basse pression. La rotation est de sens horaire dans l'hémisphère sud et de sens anti-horaire dans l'autre hémisphère. Ils se forment au-dessus des eaux chaudes des mers tropicales. Les ouragans se situent dans l'Atlantique Nord, le golfe du Mexique ainsi qu'à l'est du Pacifique nord. Les typhons sont situés dans l'ouest du Pacifique nord ainsi qu'au large de la mer de Chine méridionale. Enfin, les cyclones sont situés dans l'océan Indien.

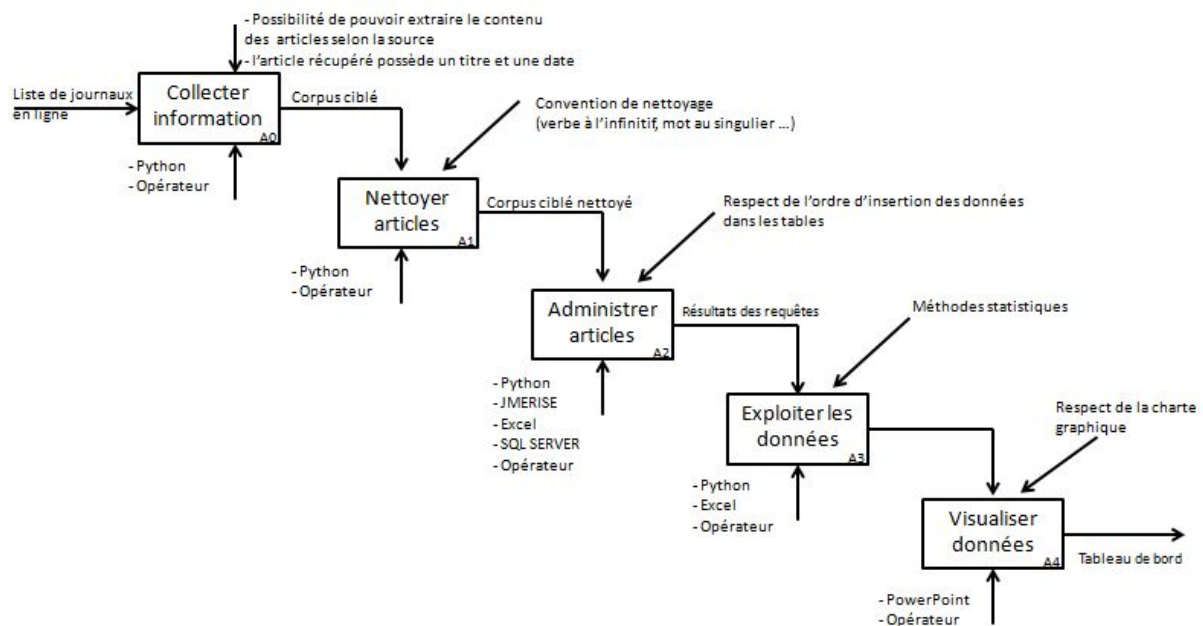
Une inondation est un débordement d'eaux qui recouvre une étendue de terrain.

Un séisme / tremblement de terre est dû à la fracturation des roches en profondeur dans la croûte terrestre. Cette fracturation libère de l'énergie qui se traduit à la surface par des vibrations du sol. La plupart des séismes se produisent au niveau des plaques tectoniques mais il arrive aussi qu'il y en ait à l'intérieur des plaques (séismes intraplaques).

## 4. Gestion de projet

### 4.1. Organisation du travail

Au début du projet nous avons réalisé un diagramme SADT (en anglais Structured Analysis and Design Technique), ou méthode d'analyse fonctionnelle descendante, est une méthode graphique qui part du général pour aller au particulier.



Elle permet de décrire des systèmes complexes où coexistent différents flux de matière d'œuvre. Il nous a permis de poser les différentes étapes de notre processus, les outils et méthodes que nous allons utiliser.

Nous avons également conçu un diagramme de GANTT afin de définir les principales tâches nécessaires à la réalisation du processus. Nous nous sommes également répartis différents rôles au sein de l'équipe afin d'assurer une cohésion. Noémie Deloeuvre est la chef du groupe, Killian Buhr est chargé de la gestion de configuration et Nicolas Brouté est chargé de la qualité.



Notre schéma prévisionnel nous a permis d'avoir une vision global du projet :

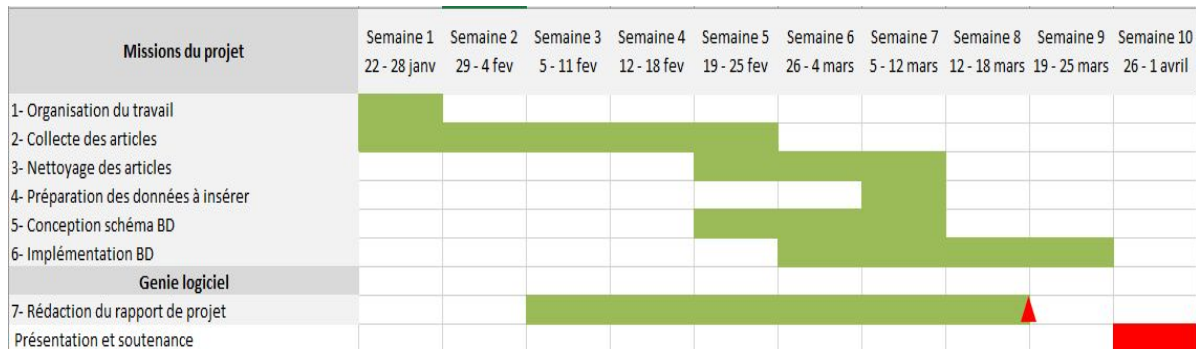
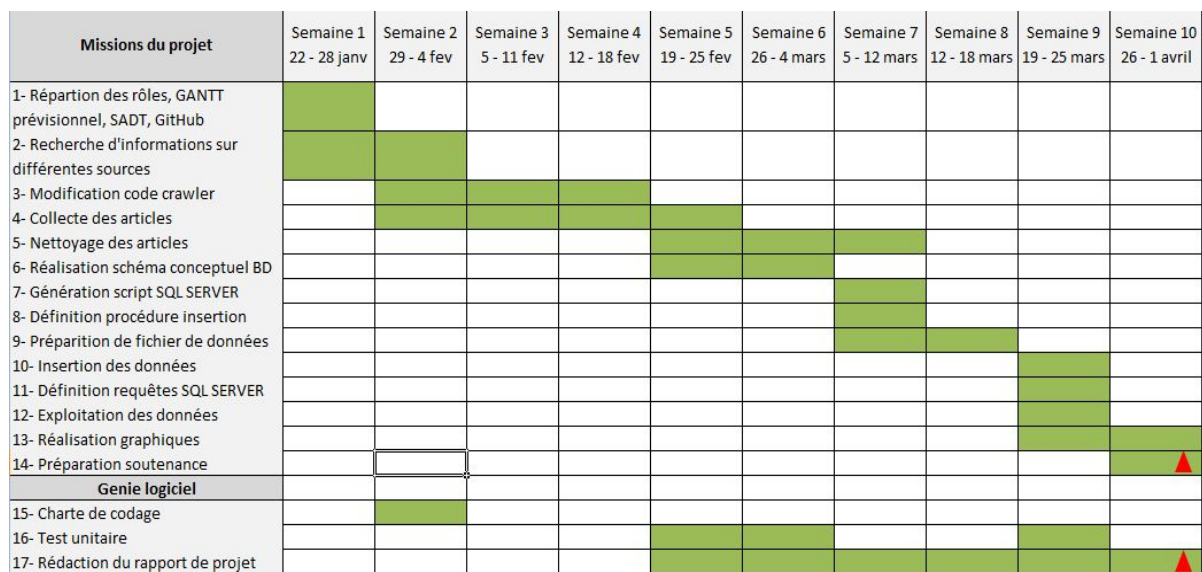


Diagramme de GANTT final :



En comparant nos diagrammes de GANTT au début et à la fin du projet, nous pouvons dire que nous avons été trop succincts dans la définition des missions ainsi que leurs durées.

Nous avons dépassé la date limite concernant la collecte des articles. En effet, sur certaines sources, nous avons l'obligation d'avoir un temps d'attente entre la collecte d'articles afin que le site ne nous bloque pas.

Nous avons pris du temps pour la création des fichiers de données nécessaire à l'insertion dans les tables de la base de données.

Nous n'avons pas rencontré de problème particulier pour la partie visualisation.



## 4.2. Outils utilisés

Afin de nous coordonner nous avons utilisé GitHub qui est une plateforme open source de gestion de versions et de collaboration. Elle repose sur Git, un système de gestion de code open. Git permet de stocker le code source d'un projet et de suivre l'historique complet de toutes les modifications apportées à ce code. GitHub facilite la programmation collaborative.

Nous avons utilisé le logiciel JMerise afin de créer le Modèle de Conception de données (MCD). A partir de celui-ci, le logiciel élabore également le Modèle Logique de Données (MLD) ainsi que le script de création des tables adapté à SQL Server.

Nous avons utilisé SQL Server comme système de gestion de base de données relationnelle.

Python est un langage de programmation de haut niveau d'abstraction qui favorise la programmation impérative structurée et orientée objet. Nous avons utilisé la version 3.6 de Python. Python fut notre principal outil pour développer nos fonctions. Nous avons utilisé différents modules tels que numpy, spacy, wordcloud, matplotlib ou d'autres encore visibles dans le cahier de recette.

Le dernier outil que nous avons utilisé est le logiciel R. C'est un langage de programmation et un logiciel libre dédié aux statistiques et à la science des données. Il nous a permis de concevoir un modèle de régression linéaire ainsi que de réaliser des tests statistiques.

## 5. Démarche de projet

### 5.1. Recherche d'informations et préparation de données

La recherche d'informations est l'étape qui permet d'identifier celle qui est utile dans les sources sélectionnées pour alimenter l'analyse. La collecte de données se fera selon le périmètre défini dans l'introduction.

#### 5.1.1. Recherche de données

Chaque année la formation organise un projet inter-promo. Cette année l'objectif était de réaliser un site web pour afficher la tendance des mots dans les articles selon plusieurs sources.

Grâce à ce projet nous avons déjà une liste de journaux définie. Nous avons choisi de traiter quelques journaux afin d'avoir le temps d'exploiter les données. Nous avons sélectionné les sources suivantes : Nouvel Obs, Le Point, Le Monde, Le Figaro et Libération.

Ensuite, chacun des membres a effectué des recherches avec les mots clés définissant notre sujet. Nous avons remarqué que certaines sources ne disposait pas d'un historique assez conséquent. Par exemple, le journal Libération ne possède pas d'historique avant l'année 2008, or nous avons besoin de récolter les articles issus des années 2000. Un autre exemple est celui du journal Futura-Science qui n'affiche pas le thème en fonction des articles (ex: cyclone) sur sa page HTML, ainsi les données n'avaient pas de sens d'être récupérées car toute l'étude se base sur les mots et les thèmes.

#### 5.1.2. Collecte de données

Notre première priorité a été de récupérer les articles pour pouvoir avancer sur la conception de fonction de nettoyage et la préparation des données. Grâce au projet inter promo, des crawlers (robot logiciel en charge de l'exploration des sites et contenus Internet) pour différentes sources ont été conçu. Nous avons donc pu ré-utiliser ces codes disponibles. Nous avons dû les adapter afin qu'ils récupèrent les articles selon la catastrophe recherchée.

Les crawler s'appuient sur un fichier "utils" créé lors du projet inter-promo permettant d'avoir des identifiants uniques pour chaque article, de mettre les articles en format json et de les exporter dans ce même format. Ce code permet également de ne pas récupérer deux fois le même article pour un même journal à l'aide d'une table de hachage. Une fonction permet également de parser un url afin de pouvoir analyser son contenu.

Etant donné que les codes réalisant les crawlers ont été réalisés par différentes personnes, leurs "mise en place" peut varier d'un code à l'autre. Un crawler est généralement composé de trois fonctions :

- Une fonction permet de récupérer pour un article donné toutes les informations le concernant afin de les mettre en format json.
- Une fonction permet de récupérer toutes les url des articles qui nous intéressent sur le site internet du journal.
- Une dernière fonction fait appel aux deux précédentes afin de mettre en forme les articles. Cette fonction permet également de faire appel au fichier "utils" afin d'exporter tous les articles en fichier json.

Nous avons décidé de récupérer les articles pour Libération même si l'historique n'allait pas au delà de 2008. Pour certaines sources nous n'avons pas pu récupérer toutes les catastrophes naturelles. En effet la quantité d'articles était trop importante et, par conséquent, la durée de récolte était de plus d'une journée.

Une des difficultés que nous avons rencontré dans cette partie a été la compréhension du code des crawlers.

A l'issue de la recherche et de la collecte d'informations, nous avons récolté 11 535 articles dont 10 732 exploitables. La deadline pour récolter les articles avait été fixé au 23 Février 2018. Nous l'avons dépassé d'une semaine afin de récupérer quelques thèmes (dont la durée de collecte n'était pas importante) pour certains journaux.

### 5.1.3. Préparation des données

La préparation des données a été séparée en 3 grandes étapes. Premièrement, le nettoyage des mots dans les corpus ciblés. Ensuite, l'exploitation de ces mots pour en ressortir de l'information (les métadonnées). Et enfin, la préparation des fichiers de données pour l'insertion dans les tables de la BD. Il est à noter que toute cette partie a été réalisée sous Linux à cause de la librairie Spacy qui ne fonctionne pas facilement sur Windows. Les données sont donc ressorties en format csv pour être insérées sur SQL Serveur via un ordinateur sous Windows.

Exploiter tous les mots des articles n'ayant pas de sens, nous avons décidé de supprimer toutes les ponctuations ainsi que les mots vides de sens comme les conjonctions de coordinations. Nous avons repris le fichier "stopwords" utilisé par le groupe de filtrage lors du projet inter-promo pour encore diminuer les mots inutiles présents dans les textes. Nous choisissons également de mettre les verbes à l'infinitif. Les mots "est" et "était" sont des conjugaisons différentes du verbe "être", ils seront donc enregistrés comme étant un seul et même mot, le mot "être". On appelle cette technique la lemmatisation. Cette fonctionnalité est possible grâce à la librairie "Spacy".

Nous avons réalisé un traitement bien particulier pour les chiffres. En effet, les chiffres sont de très bon indicateurs pour des études comme la nôtre mais ils sont aussi très difficiles à extraire avec exactitude. Le but était donc de créer une fonction ressortant le mieux possible à quels mots correspondent les différents chiffres des textes. Une première recherche permet de détecter les mots tels que "million" ou "mille" collés aux nombres. Ceci a permis de transformer, par exemple, le nombre 2.2 millions en 2200000. Cette transformation permettra plus tard de pouvoir exploiter les chiffres. Pour chaque nombre ainsi obtenu, nous avons réussi à savoir s'il était associé à un mot en particulier. Pour ce faire nous nous sommes basés sur une liste de mots qui nous intéressait (mort, euros,...). Cela nous permettra par la suite de pouvoir exploiter les chiffres tout en sachant à quoi ils font référence.

Spacy permet de ressortir d'autres informations telles que la nature du mot (nom, verbe, adjectif, etc..) ainsi que les entités nommées qui sont au nombre de 4 :

LOC : Localisation

PERS : Personne

ORG : Organisation

PAYS : Pays

NULL : Ne fait partie d'aucun des 4 groupes précédents

Un autre moyen d'obtenir des métadonnées est la technique des TF-IDF. Cette mesure statistique permet d'évaluer l'importance d'un terme contenu dans un document, relativement à l'ensemble des articles. Nous en avons créé deux tableaux TF-IDF distincts, un pour le titre et l'autre pour le contenu des articles. Il sont tous les deux paramétrés de deux façons différentes. Pour le contenu, nous ne prenons en compte que les mots présents au moins 30 fois dans l'ensemble des articles et uniquement les 10 000 mots les plus importants. Concernant les titres, nous gardons les mots avec minimum cinq occurrences et uniquement les mille mots les plus importants. Ainsi, nous allons faire l'étude sur un total de 6 528 mots.

La dernière étape est le regroupement et l'homogénéisation de toutes les données extraites et manipulées précédemment. En effet, nous avons décidé de générer les tableaux relatifs aux tables de la BD directement dans le programme python. Ainsi, nous parcourons les différents tableaux pour faire correspondre les clés étrangères et vérifier qu'il n'existerait pas de doublons.

Ces tableaux sont exportés en csv, chaque fichier représentant une table de la BD.

L'étape de la collecte ainsi que la transformation des données nous aura pris une grande partie du projet. Nous avons pris le temps de bien réfléchir à la simplification du traitement des articles. Nous avons également essayé de définir un ensemble de questions auxquelles nous pourrions répondre grâce aux données insérées dans la BD.

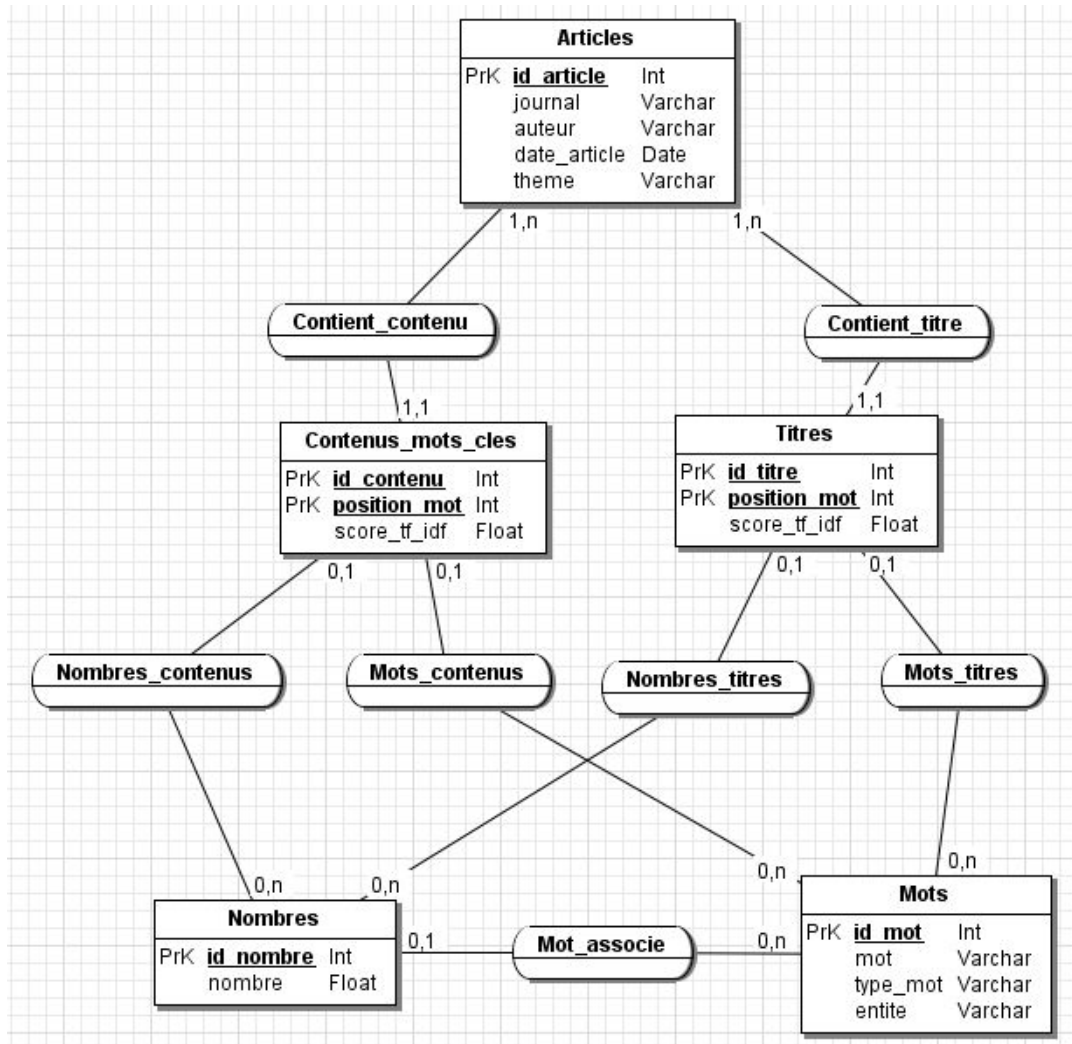
## 5.2. Administrer

Dans cette partie nous allons commencer tout d'abord par la conception du schéma conceptuel de la base de données. Ensuite nous développerons l'insertion des données triées dans celle-ci grâce à des procédures stockées. Enfin nous détaillerons les requêtes mises en place pour obtenir les données pertinentes à notre analyse.

### 5.2.1. Conception du schéma

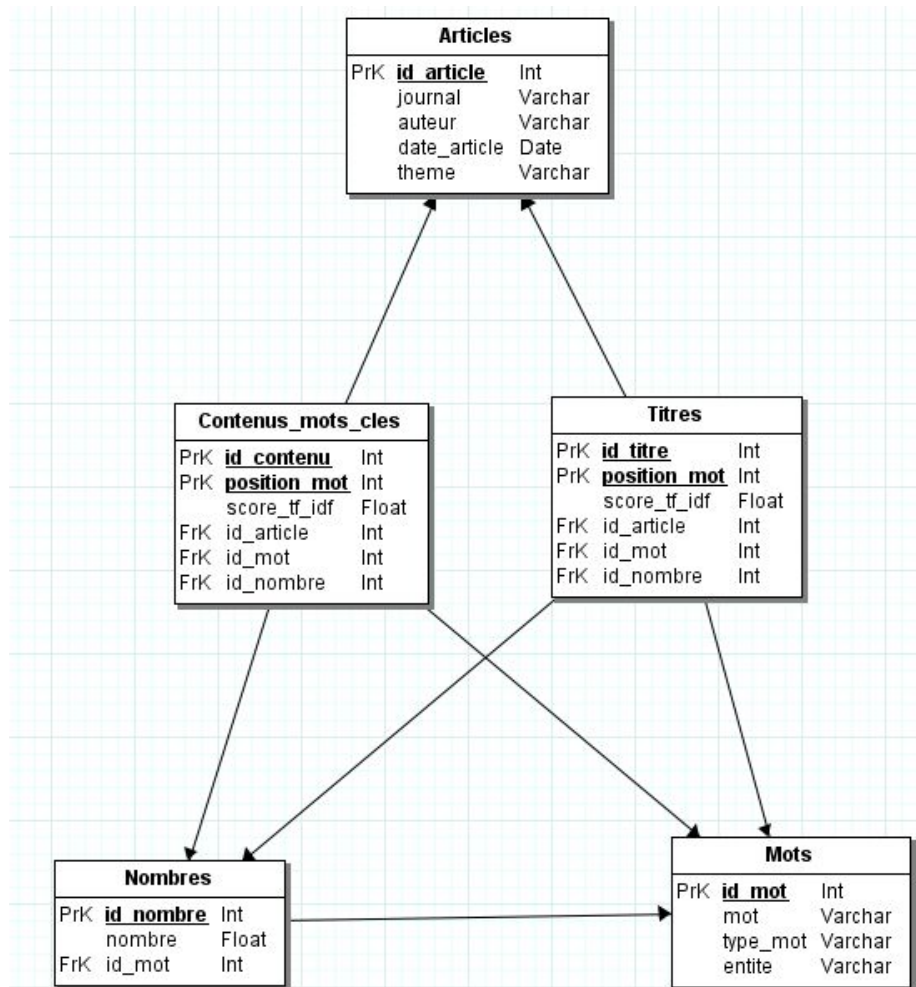
Nous avons réalisé le schéma entité association grâce au logiciel Jmerise. C'est un logiciel dédié à la modélisation des modèles conceptuels de données (MCD) pour Merise il permet la généralisation et la spécialisation des entités, la création des relations et des cardinalités ainsi que la généralisation des modèles logiques de données (MLD) et des script SQL.

## Schéma MCD





## Schéma MLD



Pour chaque article, on garde dans une table “Articles”, son identifiant, le nom du journal associé, l’auteur, la date de l’article ainsi que son thème.

Exemple de tuple : (1, “LePoint”, “LaRédaction”, “2014-12-16”, “inondation”)

Chaque tuple de la table “Titres” correspond à un mot clé d’un titre d’un article. On lui associe un identifiant de titre, sa position dans le titre, éventuellement son score tf\*idf, l’identifiant de l’article auquel le titre correspond et l’identifiant du mot associé ou du nombre associé (chaque “mot” clé peut être associé soit à un mot soit à un nombre).

Exemple de tuple si le mot est numérique (1, 1, NULL, 1, NULL, 3)

Exemple de tuple si le mot est alphanumérique: (1, 1, 0.47896, 1, 1, NULL)

Chaque tuple de la table “Contenus\_mots\_cles” correspond à un mot clé d’un contenu d’un article. On lui associe un identifiant de contenu, sa position dans le contenu, éventuellement son score tf\*idf, l’identifiant de l’article auquel il est associé, l’identifiant correspondant au

mot associé ou l'identifiant correspondant au nombre associé (chaque "mot" clé peut être associé soit à un mot soit à un nombre).

Exemple de tuple si le mot est numérique (1, 1, NULL, 1, NULL, 3)

Exemple de tuple si le mot est alphanumérique: (1, 1, 0.47896, 1, 1, NULL)

La table "Nombres" contient tous les différents nombres présents dans les différents contenus. A chaque nombre, on peut lui associer un mot. Par exemple le nombre 20 peut être associé au mot "mort". Un nombre peut être présents plusieurs fois dans la table s'il peut être associé à différents mots (ex : 20 morts et 20 euros).

Exemple de tuple : (1, 138, NULL)

La table "Mots" contient tous les mots présents dans tous les documents de façon unique. Pour chaque tuple on indique un identifiant, le mot auquel il correspond, le type de mot (verbe, adjectif,...) ainsi que le type de l'entité nommée si c'en est une.

Exemple de tuple : (26, "eau", "ADV", NULL)

### 5.2.2. Insertion des données

Afin d'insérer les données dans la base de données SQL Server, nous avons créé une procédure pour chacune des tables. Chaque procédure récupère les éléments nécessaires à l'insertion d'une ligne dans la table associée puis insère la ligne dans la table.

En langage python, nous avons créé pour chacune des tables un fichier csv contenant toutes les lignes à insérer dans la table. A l'aide d'un package sous python nous avons pu réaliser une connexion avec la base de donnée présente sur SQL Server. Nous avons directement fait appel aux procédures sur le logiciel Spyder pour insérer les données dans la base. Ceci a permis de rentrer les tuples dans chaque table de manière automatique.

### 5.2.3. Interrogation des données

Nous avons défini des questions auxquelles nous allons pouvoir répondre en interrogeant notre base données.

Chaque requête a été lancée à partir de Spyder afin de pouvoir faire différents traitements dessus et d'exporter les résultats en fichier csv de manière simple et automatique.

Dans un premier temps, nous avons commencé par récupérer des informations générales telles que le nombre d'articles par journal, par thème (inondation, cyclone, ouragan, typhon, séisme et tremblement de terre), par année ainsi que par thème et année à la fois. Nous avons également regardé les effectifs des différents mots qui qualifient les nombres (mourir, euro,...). Voir l'annexe 1 pour visualiser les différentes requêtes effectuées.

Dans un second temps nous avons réalisé des requêtes plus spécifiques afin de pouvoir répondre aux questions que nous nous posons. Ainsi nous avons fait des requêtes pouvant répondre aux questions suivantes : (Voir Annexe 2)

- Quelles sont les pays les plus touchés par les catastrophes ?
- Quels sont les verbes / adjectifs / noms les plus utilisés dans les articles et les titres ?
- Quelle est l'évolution du nombre d'articles par année par catastrophe naturelle ?

Dans un troisième temps nous avons étudié les informations que les chiffres pouvaient nous donner. Ainsi nous avons pu répondre aux questions suivantes : (Voir Annexe 3)

- Nombre moyen de mort par type de catastrophe naturelle
- Nombre moyen de mort par année
- Nombre moyen de mort par type de catastrophe naturelle et par année
- Coût moyen par type de catastrophe naturelle
- Coût moyen par année
- Coût moyen par type de catastrophe naturelle et par année

Concernant les nombres, il y avait parfois des nombres aberrants ; ainsi nous avons décidé de garder pour les morts tous les nombres inférieurs à 50 000 et pour les coûts tous les nombres inférieurs à 100 milliards.

Enfin nous avons décidé de réaliser une carte afin de pouvoir bien visualiser nos données. Nous avons donc écrit des requêtes afin d'obtenir pour chaque pays et pour chaque type de catastrophe le nombre moyen de morts ainsi que le nombre d'articles écrits dessus. Nous avons également obtenu les informations sans regrouper par type de catastrophe naturelle. Afin de pouvoir faire une carte, nous avons utilisé un fichier excel regroupant les coordonnées GPS de chaque pays. Ainsi, en regroupant le résultat de la requête avec ce fichier Excel nous avons pu obtenir une carte assez détaillée. (Voir Annexe 4)

## 5.3. Visualisation de données

### 5.3.1. Méthodes statistiques

Nous avons réalisé un modèle de régression linéaire afin de détecter une tendance dans le nombre d'articles par année. Nous avons conçu le modèle suivant :

$$Nb\ articles = 59.22 * année + 31.85$$

Le test de Student nous permet de vérifier la significativité des coefficients de la régression linéaire. En particulier, il permet de tester la nullité de chacun des coefficients. Nous avons défini l'hypothèse  $H_0$  : coefficient est nul et  $H_1$  : coefficient est différent de zéro.

Nous avons obtenu les résultats suivants avec le logiciel R :

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    31.85      154.03   0.207 0.838650
x              59.22       14.62   4.050 0.000831 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 349 on 17 degrees of freedom
Multiple R-squared:  0.4911,    Adjusted R-squared:  0.4612
F-statistic: 16.41 on 1 and 17 DF,  p-value: 0.0008313

```

On rejette l'hypothèse nulle si la p-value est inférieur à un risque alpha. Pour le coefficient directeur du modèle nous obtenons une p-value égale à 0.000831. Elle est inférieure au seuil alpha (5 %). On peut donc conclure que le coefficient directeur est significativement différent de zéro. En revanche l'intercept a une p-value égale à 0.838650 (supérieur à 5 %). On conclut que le coefficient est non significativement différent de zéro.

Le modèle s'écrit donc

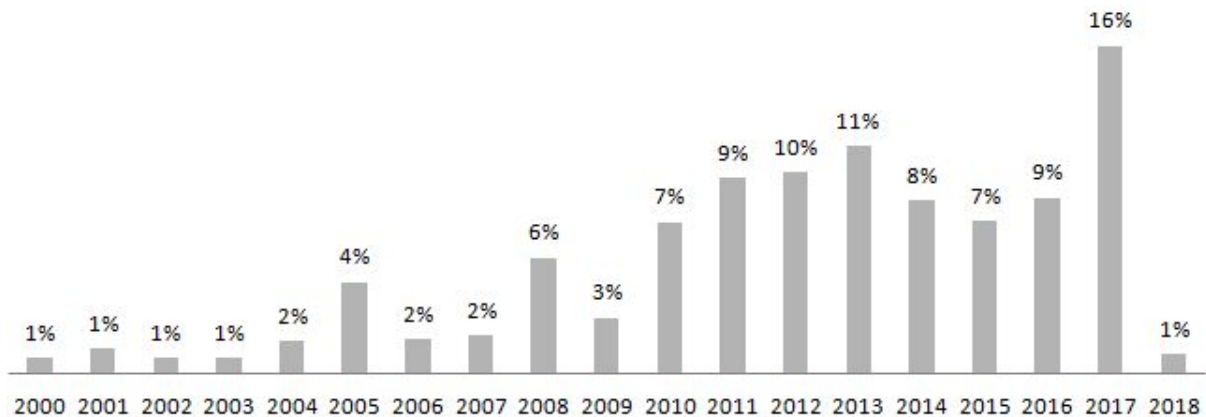
$$Nb\ articles = 59.22 * année$$

### 5.3.2. Dashboard Excel

Pour avons décidé de réaliser notre Tableau de Bord sur Excel. Il est composé de quatre onglets. Un premier onglet intitulé "Information", on y retrouve le contexte du sujet. Le second onglet intitulé "Dashboard" contient différents graphiques sur le nombre d'articles selon la catastrophe et l'année, le nombre de mort par année par tempête. Le troisième onglet de notre tableau de bord contient différents nuages de mots sur le contenu et le titre des articles. Un onglet similaire qui contient des nuages de pays les plus touchés selon la catastrophe. Enfin le dernier onglet contient des liens qui nous dirigent vers des liens HTML qui représentent des cartes des catastrophes.

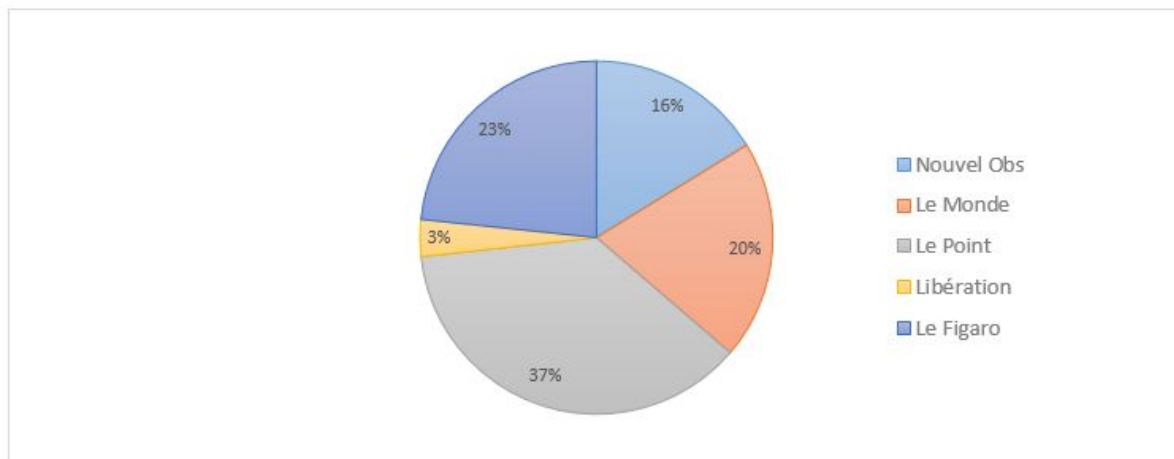
Les deux graphiques que l'on aperçoit en premier sur le Dashboard concernent la répartition des articles récoltés par année et la répartition des articles par source.

### Répartition du nombre d'articles par année



On peut observer sur le diagramme que c'est pour l'année 2017 que nous avons réussi à récupérer le plus d'article (16 %). Ensuite c'est l'année 2013, 2012 et 2011 avec respectivement 11 %, 10 % et 9%. Nous avons réussi à récolter peu d'articles des débuts années 2000 (environ 1 à 2%).

### Répartition des articles par année

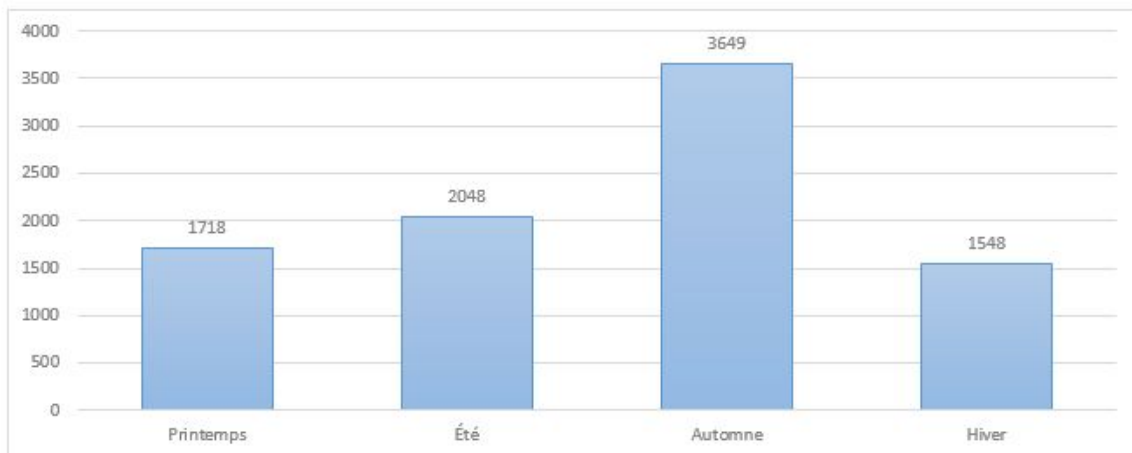


Le diagramme circulaire nous indique qu'il y a 37 % des articles qui proviennent du journal Le Point. Il y en a 23 % provenant du Figaro. Les articles du journal Libération représentent uniquement 3 % des articles récupérés.



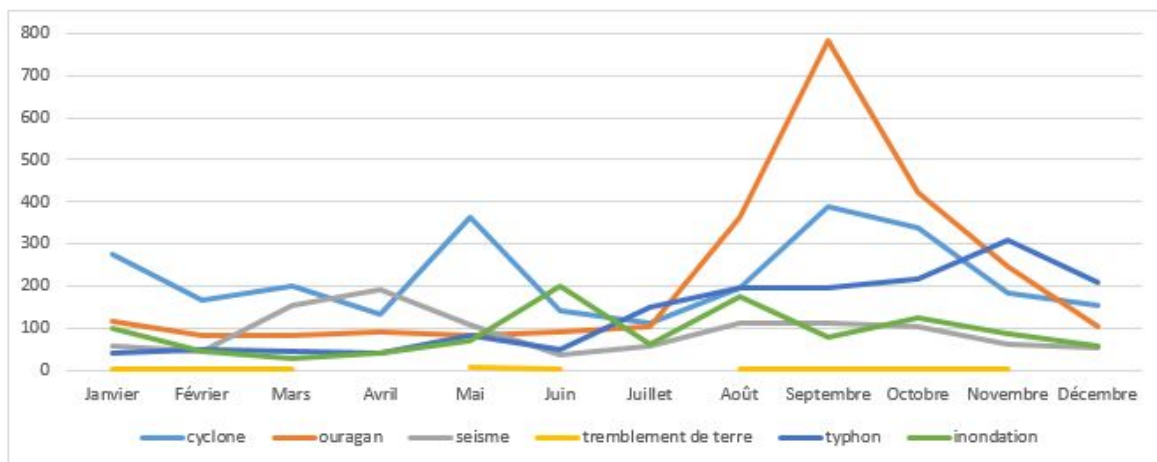
Ensuite nous avons voulu montrer à travers deux graphiques l'évolution des articles selon la saison, puis par catastrophe par mois.

### Nombre d'articles par saison



On remarque qu'il y a beaucoup d'articles qui paraissent au cours de l'automne (3649). C'est en hiver que l'on possède le moins d'articles (seulement 1548).

### Nombre d'articles par thème et par mois



On peut observer sur ce graphique le nombre d'articles par mois selon la catastrophe. Comme l'a montré le graphique précédent, c'est en automne qu'on a le plus d'articles sur les catastrophes. On observe une très forte augmentation des ouragans (800 articles). Ceci concorde avec la saison cyclonique dans l'Atlantique nord selon la définition de l'Organisation météorologique. Il y a une augmentation moins importante des cyclones (un peu moins de 400 articles).



Pour terminer, nous retrouvons trois graphiques qui analysent la répartition des articles, le nombre de morts ainsi que le coût selon la catastrophe et l'année. Nous avons ajouté deux segments (un segment permet de trier les éléments de façon visuelle).

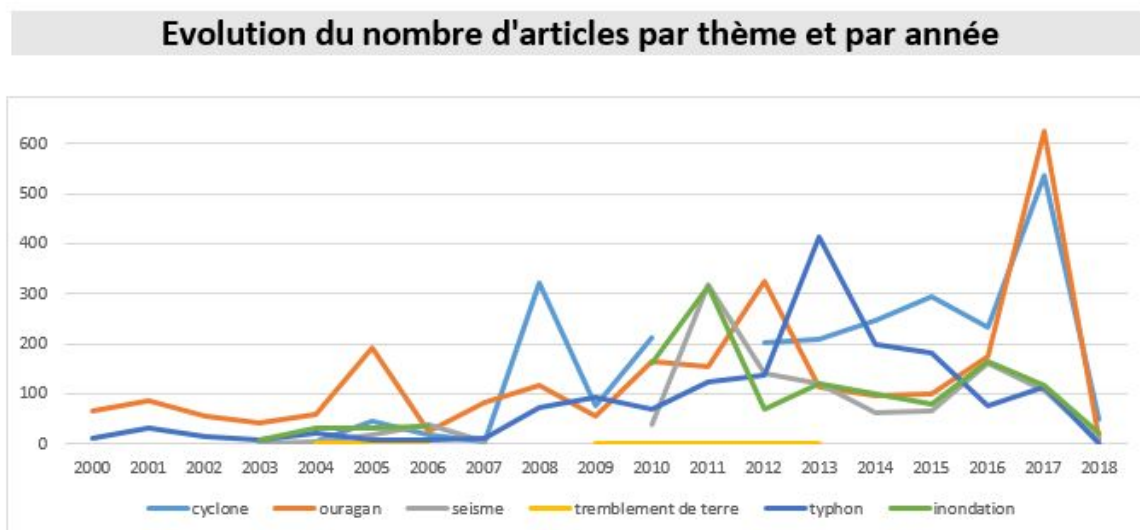
**Thème**

- cyclone
- inondation
- ouragan
- seisme
- tremblement de terre
- typhon

**Années**

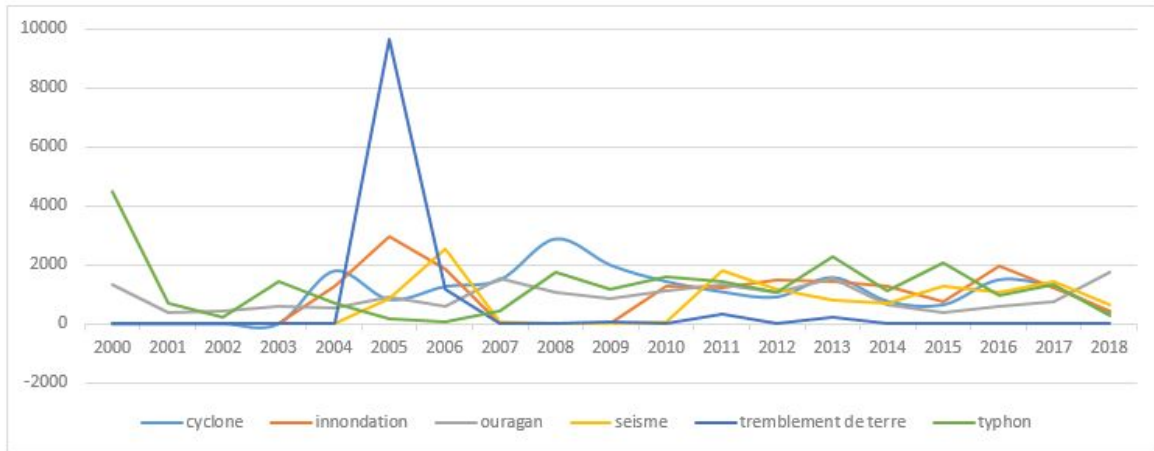
- 2000
- 2001
- 2002
- 2003
- 2004
- 2005
- 2006
- 2007

Ces deux boîtes nous permettent d'interagir dynamiquement avec les graphiques. On peut sélectionner un à plusieurs thèmes selon une à plusieurs années. Cela permet à l'utilisateur de faire une multitude de comparaisons.



On observe sur le graphique que les trois premières années du XXIe siècle, nous avons réussi à récupérer uniquement des articles portant sur les ouragans et sur les typhons. En 2017, nous avons récupéré beaucoup d'articles concernant les cyclones et les ouragans (plus de 500). Lorsque la courbe est "coupée" cela signifie que la donnée est manquante ou bien qu'il n'y pas eu d'article cette année là.

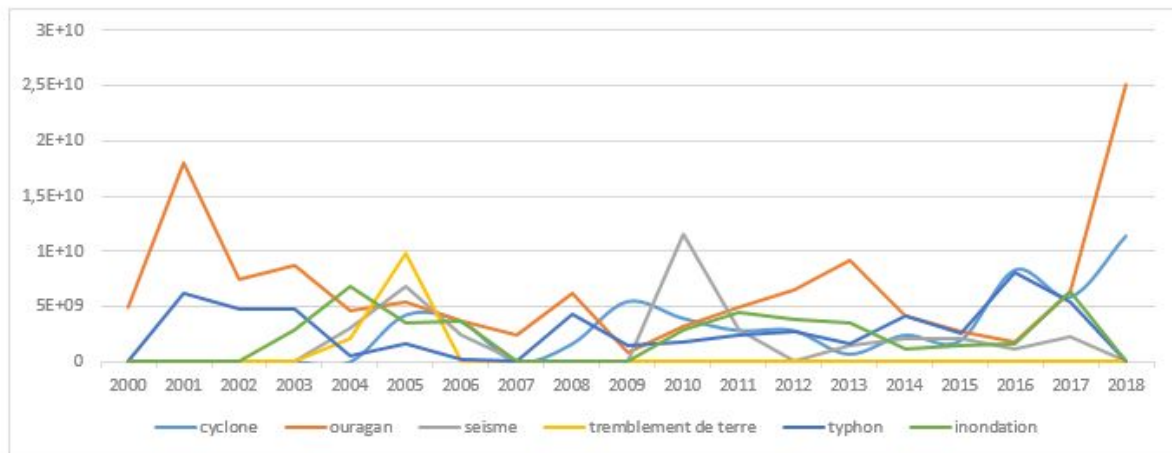
### Nombre de morts selon la catastrophe naturelle et l'année



On observe sur le graphique l'évolution de la moyenne du nombre de mort par catastrophe et par année. Le nombre moyen de morts par année et par catastrophe fluctue en 0 et 2000 morts. On remarque un pic important pour la catastrophe "tremblement de terre" en 2005. En effet cet année là, il a eu un tremblement de terre extrêmement meurtrier au Pakistan.

On trouve également un graphique sur l'évolution des coûts des catastrophes naturelles

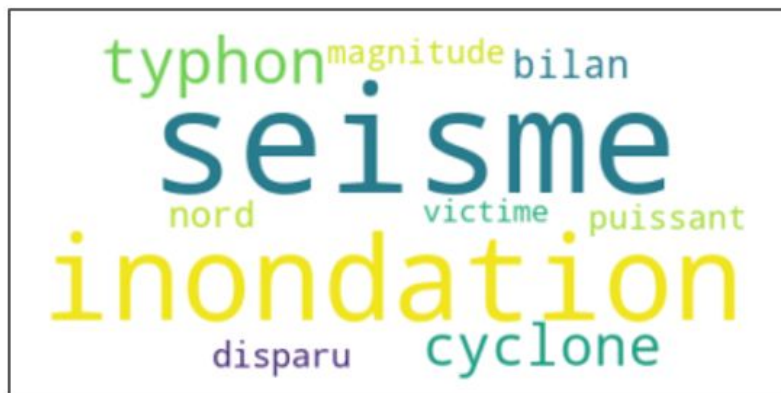
### Coût selon la catastrophe naturelle et l'année



On observe que le coût des catastrophes varie entre 0 et plus de 10 milliards d'euros (sauf la catégorie ouragan qui a dépassé les 20 milliards en 2018).

On retrouve présents dans le Dashboard (onglet intitulé WordCloud) différents nuages de mots concernant les articles

#### Top 10 des noms dans le titre des articles



On a réalisé des nuages de mots qui représentent le top dix mots selon leurs catégories (verbe, nom et adjectif) contenus dans le titre et le corps des articles. On peut observer sur le nuage de mots ci-dessous les 10 noms communs les plus utilisés dans les titres des articles. Les mots les plus importants sont “séisme” et “inondation” (taille plus importante que les autres). Arrivent ensuite les mots “cyclone” et “typhon”.

Dans notre onglet intitulé “CountriesCloud”, on y trouve des nuages de mots sur les pays les plus touchés selon une catastrophes naturelles ainsi qu’un nuage de mots sur les pays les plus touchés en général.

#### Top 10 des pays les plus touchés



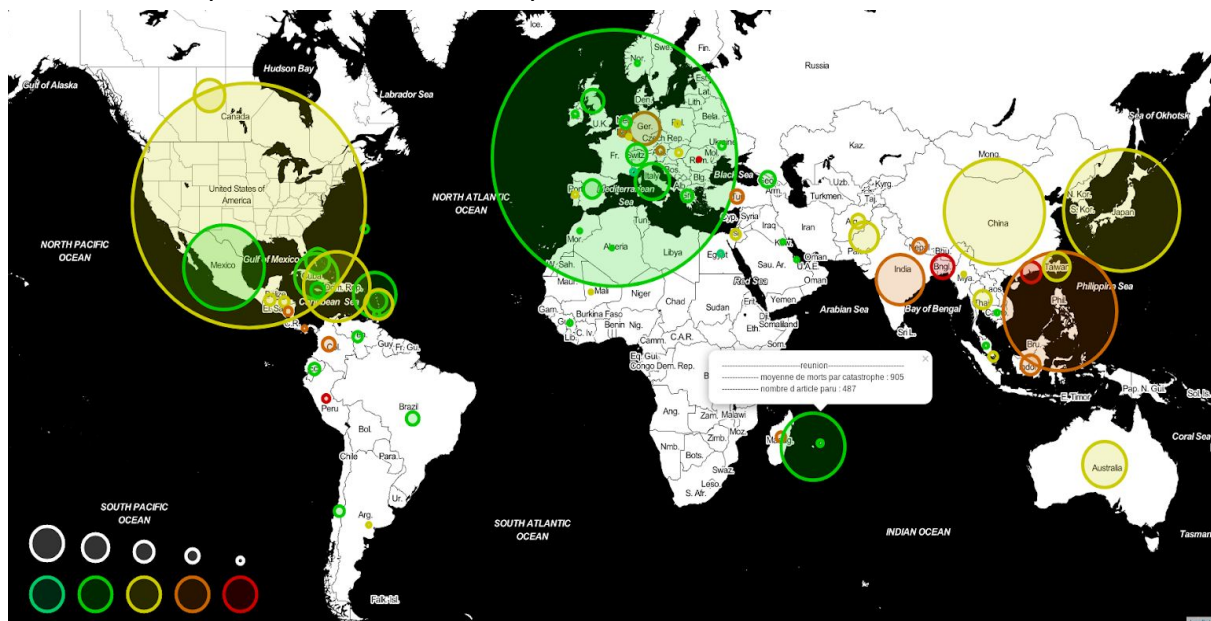
Sur le graphique des pays les plus touchés, la France et les Etats-Unis arrivent en tête. Ensuite on retrouve la Chine et le Japon, puis le Mexique, Haïti et Philippines à tailles égales. La Réunion, l'Australie et l'Inde arrivent en dernière position.

Dans le dernier onglet 'Maps', on y trouve des liens HTML qui nous dirigent vers des représentations cartographiques des catastrophes.

### 5.3.3. Carte HTML

La représentation cartographique est une représentation de l'information plus que cohérente dans notre étude. En effet, lors de cette dernière nous avons récupéré énormément de nom de ville et de pays. Ainsi, il est facile d'associer les catastrophes aux pays.

Pour cela, nous avons utilisé la librairie python "folium". Elle permet de générer des cartes en html et d'y insérer nos données. Nous avons donc généré 7 cartes. La première représente toutes les catastrophes naturelles confondues en fonction des pays cités dans les articles et par rapport à la moyenne de morts (cf. image ci-dessous ou annexe 5). Les 6 autres cartes représentent les 6 catastrophes naturelles ciblées.



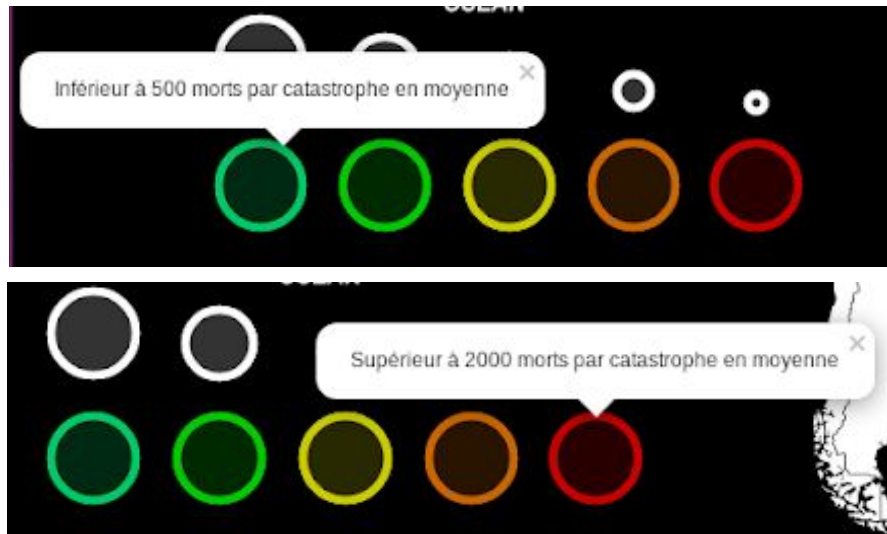
On remarque directement que les zones géographiques les plus citées dans nos articles sont :

- l'Europe de l'Ouest
- L'Amérique du nord/centrale et les caraïbes
- Madagascar et les îles proches
- Le nord de l'océan Indien
- L'est de l'Asie

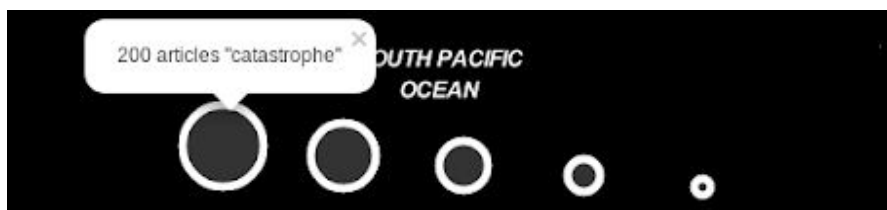
Les plus meurtrières en moyenne se situent au nord de l'océan indien et de l'est de l'Asie.

Les cartes ont toutes été créées de la même façon:

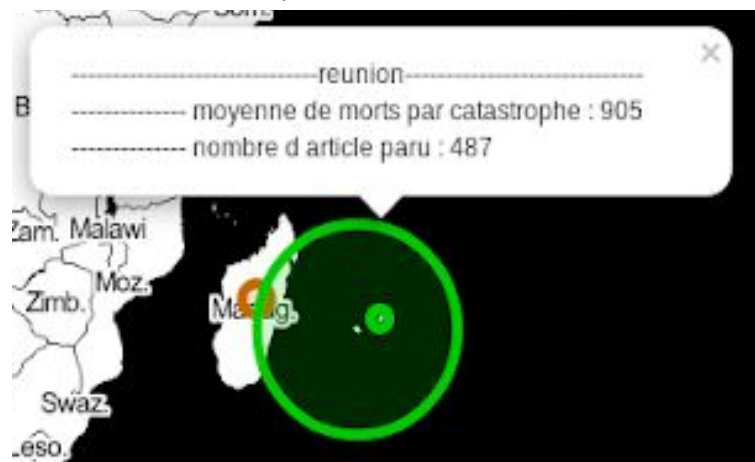
- Une légende en bas à gauche avec un code couleur représentant le nombre moyen de morts par pays et une légende .



- Une autre légende pour le nombre d'articles, représenté par la taille des cercles.



- Des infobulles pour chaque pays cité dans les articles.

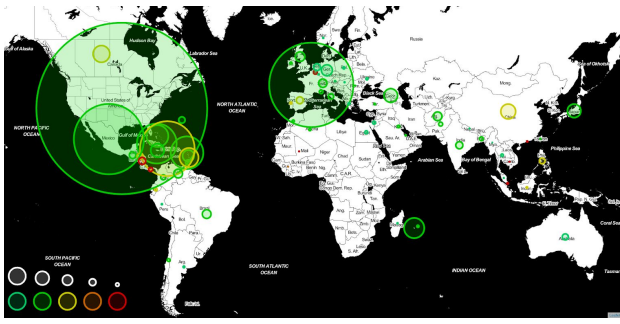
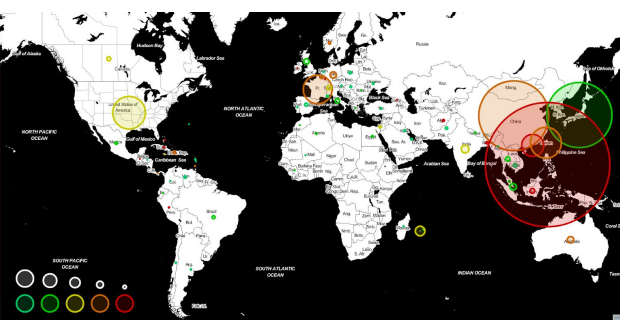
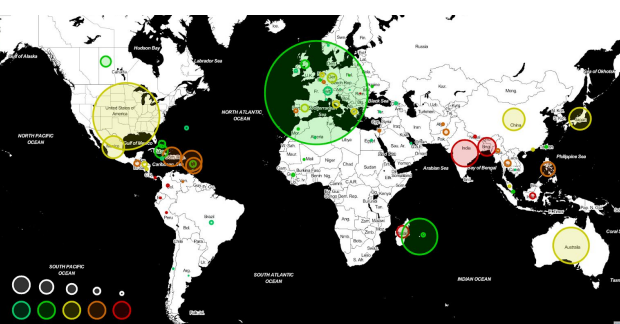




D'après les définitions (cf page 7), les ouragans, les cyclones et les typhons sont en réalité le même type de catastrophe mais localisés à différents endroits du globe (voir image ci-dessous).



Cela se voit à travers notre étude comme le montre les 3 images suivantes représentant respectivement, la carte des ouragans, des typhons et des cyclones.

<p><b>Ouragans</b></p> <p>Zone dense en Amérique centrale, du nord et dans les caraïbes. Correct par rapport aux attentes.</p>	
<p><b>Typhons</b></p> <p>Zone dense au niveau du Japon, Chine, Taiwan, Hong-kong et les Philippines. Correct par rapport aux attentes.</p>	
<p><b>Cyclones</b></p> <p>Zone peu dense. Les différentes valeurs sont réparties entre l'océan Indien, les Caraïbes et l'ouest de l'océan Pacifique. Les résultats sont moins satisfaisants que les 2 précédents.</p>	



## 6. Gestion de configuration

### 6.1. Script de création de la base de données

Nous avons utilisé le logiciel JMerise afin de réaliser les schémas de la base de données. Grâce à ce logiciel nous avons pu générer le script afin de créer la base de données sous SQL Server.

```

/*-----
*      Script SQLSERVER
-----*/

/*-----
-- Table: Articles
-----*/
CREATE TABLE Articles(
    id_article INT NOT NULL ,
    journal    VARCHAR (30) ,
    auteur     VARCHAR (50) ,
    date_article DATETIME ,
    theme      VARCHAR (25) ,
    CONSTRAINT prk_constraint_Articles PRIMARY KEY NONCLUSTERED (id_article)
);

/*-----
-- Table: Titres
-----*/
CREATE TABLE Titres(
    id_titre   INT NOT NULL ,
    position_mot INT NOT NULL ,
    score_tf_idf FLOAT ,
    id_article INT NOT NULL ,
    id_mot     INT ,
    id_nombre  INT ,
    CONSTRAINT prk_constraint_Titres PRIMARY KEY NONCLUSTERED
(id_titre,position_mot)
);

```

```

/*-----
-- Table: Contenus_mots_cles
-----*/
CREATE TABLE Contenus_mots_cles(
    id_contenu INT NOT NULL ,
    position_mot INT NOT NULL ,
    score_tf_idf FLOAT ,
    id_article INT NOT NULL ,
    id_mot INT ,
    id_nombre INT ,
    CONSTRAINT prk_constraint_Contenus_mots_cles PRIMARY KEY
NONCLUSTERED (id_contenu,position_mot)
);

```

```

/*-----
-- Table: Mots
-----*/
CREATE TABLE Mots(
    id_mot INT NOT NULL ,
    mot VARCHAR (30) NOT NULL ,
    type_mot VARCHAR (25) ,
    entite VARCHAR (40) ,
    CONSTRAINT prk_constraint_Mots PRIMARY KEY NONCLUSTERED (id_mot)
);

```

```

/*-----
-- Table: Nombres
-----*/
CREATE TABLE Nombres(
    id_nombre INT NOT NULL ,
    nombre FLOAT ,
    id_mot INT ,
    CONSTRAINT prk_constraint_Nombres PRIMARY KEY NONCLUSTERED
(id_nombre)
);

```

```

ALTER TABLE Titres ADD CONSTRAINT FK_Titres_id_article FOREIGN KEY (id_article)
REFERENCES Articles(id_article);

```

```
ALTER TABLE Titres ADD CONSTRAINT FK_Titres_id_mot FOREIGN KEY (id_mot)
REFERENCES Mots(id_mot);
ALTER TABLE Titres ADD CONSTRAINT FK_Titres_id_nombre FOREIGN KEY
(id_nombre) REFERENCES Nombres(id_nombre);
ALTER TABLE Contenus_mots_cles ADD CONSTRAINT
FK_Contenus_mots_cles_id_article FOREIGN KEY (id_article) REFERENCES
Articles(id_article);
ALTER TABLE Contenus_mots_cles ADD CONSTRAINT FK_Contenus_mots_cles_id_mot
FOREIGN KEY (id_mot) REFERENCES Mots(id_mot);
ALTER TABLE Contenus_mots_cles ADD CONSTRAINT
FK_Contenus_mots_cles_id_nombre FOREIGN KEY (id_nombre) REFERENCES
Nombres(id_nombre);
ALTER TABLE Nombres ADD CONSTRAINT FK_Nombres_id_mot FOREIGN KEY (id_mot)
REFERENCES Mots(id_mot);
```

## 6.2. Procédures

Afin d'insérer les données nous avons décidé de créer des procédures pour chacune des tables. Chaque procédure récupère les données nécessaires d'une table puis les insère dans la table correspondante. A l'aide de la librairie "pyodbc" on réalise une connexion avec la base de données présente dans SQL Server puis on fait appel aux procédures afin d'insérer les données présentes dans les fichiers csv réalisés au préalable.

```
CREATE PROC INSERTION_ARTICLE
    @pid_article INT,
    @pjournal VARCHAR (30),
    @pauteur VARCHAR (50),
    @pdate_article DATETIME,
    @ptheme VARCHAR (25)
AS
INSERT INTO Articles (id_article,journal,auteur, date_article, theme) VALUES(@pid_article,
@pjournal, @pauteur, @pdate_article, @ptheme);
```

Attention sous SQL Server il faut rentrer les dates sous la forme YYYY-DD-MM. Elles apparaîtront dans la table de la façon suivante : YYYY-MM-DD.

```
CREATE PROC INSERTION_TITRE
    @pid_titre INT ,
    @pposition_mot INT ,
    @pscore_tf_idf FLOAT ,
    @pid_article INT ,
```

```

        @pid_mot      INT ,
        @pid_nombre   INT
AS
INSERT INTO Titres (id_titre, position_mot, score_tf_idf, id_article, id_mot, id_nombre)
VALUES (@pid_titre, @pposition_mot,
@pscore_tf_idf,@pid_article,@pid_mot,@pid_nombre);

CREATE PROC INSERTION_CONTENU_MOTS_CLES
        @pid_contenu INT ,
        @pposition_mot INT ,
        @pscore_tf_idf FLOAT ,
        @pid_article INT ,
        @pid_mot      INT ,
        @pid_nombre INT
AS
INSERT INTO Contenus_mots_cles (id_contenu, position_mot, score_tf_idf, id_article,
id_mot, id_nombre) VALUES (@pid_contenu, @pposition_mot, @pscore_tf_idf,
@pid_article, @pid_mot, @pid_nombre);

CREATE PROC INSERTION_MOTS
        @pid_mot INT ,
        @pmot     VARCHAR (30) ,
        @ptype_mot VARCHAR (25) ,
        @pentite  VARCHAR (40)
AS
INSERT INTO Mots (id_mot, mot,type_mot,entite) VALUES
(@pid_mot,@pmot,@ptype_mot,@pentite);

CREATE PROC INSERTION_NOMBRES
        @pid_nombre INT ,
        @pnombre FLOAT ,
        @pid_mot INT
AS
INSERT INTO Nombres (id_nombre, nombre, id_mot) VALUES (@pid_nombre, @pnombre,
@pid_mot);

```

## 7. Assurance et contrôle qualité

Afin d'assurer la meilleure compréhension du code possible nous avons créé une charte de codage. Celle-ci a pour but d'homogénéiser la présentation du code et d'avoir la meilleure lisibilité possible.

Dans le but d'avancer le plus efficacement possible, nous avons régulièrement fait des revues. Dans chacune d'entre elles nous établissions où nous étions par rapport au schéma de Gantt établi.

### 7.1. Charte de codage

Les conventions de codage visent essentiellement à améliorer la lisibilité du code : elles doivent permettre d'identifier du premier coup d'oeil un maximum de choses dans le code, de se repérer facilement et de savoir où trouver les choses.

#### 7.1.1. Convention de nommage des fichiers

- Chaque journal possède son propre dossier contenant les articles.
- Le nom du dossier correspond au nom du journal (ex journal : Le Monde à nom du dossier : LeMonde)
- Les fichiers python contenant le code doivent être en minuscules et de la forme suivante pour la collecte des données : `crawler_nomdujournal.py`

#### 7.1.2. Convention de nommage des variables et fonctions

- Les noms des variables et des fonctions doivent être en anglais
- Les variables et les fonctions doivent être écrites en minuscule
- Les variables doivent avoir des noms clairs (ne pas appeler une variable « a » sauf pour les boucles)
- Si les noms des variables ou des fonctions contiennent plusieurs mots, les séparer à l'aide d'underscore « \_ »

#### 7.1.3. Commentaires du code

- Les commentaires doivent être écrits en français.
- Toutes vos fonctions doivent être commentées, de façon à indiquer ce que prend la fonction en entrée et ce qu'elle retourne, suivie d'un petit résumé de ce qu'elle effectue.
- Indiquez dans les commentaires ce que fait le code, pas comment il le fait.

#### 7.1.4. Partie codage

- Chaque code doit être encodé en utf-8.
- Le code doit être correctement indenté
- Le code ne doit pas être compact, pour une meilleure visibilité faire des sauts à la ligne
- Veillez à toujours bien espacer vos variables de vos opérateurs.
- Une ligne de code ne doit pas excéder 80 caractères
- Toutes les variables et tous les packages déclarés doivent être utilisés
- Ne déclarez pas une variable qui n'est utilisée qu'une seule fois.
- Pour les fonctions faire attention que le seul « return » ne se situe pas dans un « if » dont la condition n'est pas toujours vérifiée

Pour vérifier une grande partie des points ci-dessus, dans Spyder allez dans Outils → Préférences → Editeur → Introspection et analyse de code. Cochez la case à côté de « Analyse de code en temps réel dans l'éditeur (PEP8) ».

#### 7.2. Revues

- Revue du 02/02/2018

Nous avons choisi comme sujet 'L'évolution des catastrophes naturelles au XXIe siècle'. Les professeurs encadrants ont validé le sujet. Nous voulons observer à travers les articles, si les catastrophes augmentent, leurs puissances ainsi que le nombre de morts.

Nous allons définir une liste de journal qui possède un historique suffisant selon un périmètre défini pour récolter les articles. Nous allons réaliser également un GANTT prévisionnel ainsi qu'un SADT pour définir le processus

- Revue du 19/02/2018

Nous avons presque fini de collecter les articles, nous serons en retard vis-à-vis de la deadline pour la collecte. Nous avons commencé en parallèle à nettoyer les articles (supprimer les mots vides) et à concevoir une fonction qui calcule le TF IDF de chaque mot pour l'ensemble de notre collection.

Nous avons présenté un schéma conceptuel de notre base de donnée à Monsieur Mokadem qu'il a validé.

Il nous reste à finir de récolter les articles, réfléchir à comment exploiter les nombres (de morts, dégâts) dans les articles pour nos analyses. Nous devons également réaliser



l'implémentation de la base de données. Nous devons réfléchir à comment insérer les données.

- Revue du 09/03/2018

Nous avons parlé de notre avancement avec Madame Bashoun. A cette date, nous avons fini la collecte des articles (11 535 articles). L'étape de nettoyage était presque finie. Nous sommes capable de supprimer les mots vides, calculer le score TF IDF de chaque mot et de repérer les entités (pays, organisation). Il reste quelques petits réglages à réaliser pour préparer la bonne mise en forme des données pour les insérer dans la base de données.

Du côté de la base de données, le schéma est validé et nous avons généré le script pour SQL SERVER.

Il nous reste à définir les procédures stockées pour insérer automatiquement les données depuis Spyder. Ensuite nous pourrons faire des requêtes pour obtenir les données nécessaires à nos analyses.

- Revue du 19/03/2018

Nous avons fait un point avec les encadrants du projet. Les données ont été insérées dans la base de données. Quelques vérifications doivent encore être faites pour être sûr qu'il n'y ait pas eu d'erreurs lors de l'insertion des données.

Il nous reste à créer des requêtes afin de pouvoir visualiser nos données et ainsi obtenir des résultats que nous montrerons lors de la soutenance.

- Revue du 26/03/2018

Nous avons fait un point avec Madame Bashoun et Monsieur Mokadem pour vérifier que les données étaient bien insérées dans la base de données et que nous pouvons faire des requêtes. Pour vérifier cela, nous avons lancé quelques requêtes dans SQL Server pour démontrer que tout fonctionnait.

Il nous reste à finaliser le rapport et le cahier de recettes. Le diaporama pour la soutenance va être commencé sous peu.

### 7.3. Tests

Les tests ont pour objectif principal d'identifier un nombre maximum de comportements problématiques du programme afin d'en augmenter la qualité. Dans cette optique, nous avons effectué 2 tests sur notre code de nettoyage.

Premièrement, un test système. Il a eu pour but de tester la robustesse du code. Pour cela, nous avons testé ce dernier avec des données erronées. Ainsi nous avons pu rajouter des contraintes dans le code afin d'améliorer sa robustesse. Concernant les performances du code, nous avons réussi à passer de 40h à 8h de traitement pour l'intégralité du nettoyage.

Ensuite, nous avons testé un test dynamique pour la classification selon le niveau d'exécution. Pour effectuer ce test nous avons sélectionné les 100 premiers articles des 5 sources que nous possédons, soit un corpus de 500 articles à la place de 11 535. Ainsi, nous avons pu voir rapidement si certaines sources avaient des spécificités que nous n'avions pas pris en compte. Un exemple serait « Le Monde » qui avait un format de date différents des autres sources. Nous avons donc adapté le code en conséquence.

## 8. Bilan

### 8.1. Bilan client

Au travers de ce projet nous avons voulu essayer de montrer que le nombre et la puissance des catastrophes naturelles augmentait au cours du XXI<sup>e</sup> siècle. Afin d'observer les résultats Il est fourni au client un tableau de bord avec l'étude des différentes catastrophes, les programmes de collecte d'articles, de nettoyage et de visualisation (nuage de mot et cartographie). Il est également fourni le code source nécessaire à l'implémentation de la base de données SQL SERVER.

Le client peut avoir accès à la totalité de la démarche et à la compréhension des résultats grâce au rapport de projet fourni. Un cahier de recette est également joint afin de comprendre les différents programmes implémentés.

### 8.2. Bilan Fournisseur

Du point de vue fournisseur, le cahier des charges a été respecté. En effet, nous avons réalisé les 3 grandes parties de ce projet avec succès. Les méthodes de génie logiciel ont été utilisées comme il l'était demandé pour avoir la meilleur gestion de projet possible.

En perspective d'évolution de notre projet, il serait pertinent d'améliorer certaines fonctions de notre programme comme par exemple :

- la détection d'entités nommées tel que les villes et région de chaque pays pour une visualisation encore plus précise.
- la fonction "keep\_number" qui associe à un nombre le mot auquel il fait référence.
- récolter plus d'articles pour chaque année et chaque catastrophe afin de combler les données manquantes.

## Annexe 1: Requêtes générales

*Nombre d'articles par journal :*

```
SELECT journal, count(id_article) as Nombre_articles FROM Articles
GROUP BY journal
```

*Nombre d'articles par thème :*

```
SELECT theme, count(id_article) as Nombre_articles FROM Articles
WHERE theme = 'seisme' or theme = 'ouragan' or theme = 'inondation'
or theme = 'tremblement de terre' or theme = 'cyclone'
or theme = 'typhon'
GROUP BY theme;
```

*Nombre d'articles par année :*

```
SELECT year(date_article) as Annee, count(id_article) as Nombre_articles
FROM Articles
GROUP BY year(date_article)
ORDER BY year(date_article)
```

*Nombre d'articles par thème et par année :*

```
SELECT theme, year(date_article) as Annee, count(id_article) as Nombre_article
FROM Articles
WHERE theme = 'seisme' or theme = 'ouragan' or theme = 'inondation'
or theme = 'tremblement de terre' or theme = 'cyclone'
or theme = 'typhon'
GROUP BY theme, year(date_article)
ORDER BY year(date_article);
```

*Nombre d'articles par thème et par mois :*

```
SELECT theme, month(date_article) as Mois, count(id_article) as Nombre_article
FROM Articles
WHERE theme = 'seisme' or theme = 'ouragan' or theme = 'inondation'
or theme = 'tremblement de terre' or theme = 'cyclone'
or theme = 'typhon'
GROUP BY theme, month(date_article)
ORDER BY month(date_article);
```

*Effectif des mots qualifiants les nombres :*

```
SELECT mot, COUNT(mot) FROM Nombres n, Mots m
WHERE n.id_mot = m.id_mot
GROUP BY mot
```

## Annexe 2 : Requêtes spécifiques

*Quelles sont les pays les plus touchés par les catastrophes ?*

```
SELECT score_tf_idf, mot FROM contenus_mots_cles c, mots m
WHERE type_entite = 'PAYS' and c.id_mot = m.id_mot
ORDER BY score_tf_idf DESC;
```

*Quels sont les verbes / adjectifs / noms les plus utilisés dans les articles et les titres ?*

*Pour le contenu :*

```
SELECT mot , score_tf_idf, type_mot FROM Contenus_mots_cles c, mots m
WHERE m.type_mot = 'VERB' and c.id_mot = m.id_mot
ORDER BY score_tf_idf DESC;
```

```
SELECT mot , score_tf_idf, type_mot FROM Contenus_mots_cles c, mots m
WHERE m.type_mot = 'ADJ' and c.id_mot = m.id_mot
ORDER BY score_tf_idf DESC;
```

```
SELECT mot , score_tf_idf, type_mot FROM Contenus_mots_cles c, mots m
WHERE m.type_mot = 'NOUN' and c.id_mot = m.id_mot
ORDER BY score_tf_idf DESC;
```

*Pour le titre :*

```
SELECT score_tf_idf, mot FROM Titres t, mots m
WHERE m.type_mot = 'VERB' and t.id_mot = m.id_mot
ORDER BY score_tf_idf DESC;
```

```
SELECT score_tf_idf, mot FROM Titres t, mots m
WHERE m.type_mot = 'ADJ' and t.id_mot = m.id_mot
ORDER BY score_tf_idf DESC;
```

```
SELECT score_tf_idf, mot FROM Titres t, mots m
WHERE m.type_mot = 'NOUN' and t.id_mot = m.id_mot
ORDER BY score_tf_idf DESC;
```

*Quelle est l'évolution du nombre d'articles par année par catastrophe naturelle ?*

```
SELECT theme, year(date_article) as Annee , count (id_article) as Nombre_article
FROM Articles
GROUP BY theme, year(date_article)
```

### Annexe 3 : Requêtes utilisant les nombres

*Nombre moyen de mort par type de catastrophe naturelle :*

```
SELECT a.theme, AVG(n.nombre) as Nombre_morts
FROM Nombres n, Contenus_mots_cles c, Articles a
WHERE n.id_nombre = c.id_nombre and a.id_article = c.id_article
and (n.id_mot = 139 or n.id_mot = 3431 or n.id_mot = 3991)
and (theme = 'seisme' or theme = 'ouragan' or theme = 'inondation'
or theme = 'tremblement de terre' or theme = 'cyclone'
or theme = 'typhon')
and nombre < 50000
and (nombre NOT BETWEEN 1950 and 2050)
GROUP BY a.theme
```

*Nombre moyen de morts par année :*

```
SELECT year(date_article), AVG(n.nombre) AS Nombre_morts
FROM Nombres n, Contenus_mots_cles c, Articles a
WHERE n.id_nombre = c.id_nombre and a.id_article = c.id_article
and (n.id_mot = 139 or n.id_mot = 3431 or n.id_mot = 3991)
and nombre < 50000
and (nombre NOT BETWEEN 1950 and 2050)
GROUP BY year(date_article)
ORDER BY year(date_article);
```

*Nombre moyen de morts par type de catastrophe naturelle et par année :*

```
SELECT a.theme, year(a.date_article), AVG(n.nombre) AS Nombre_morts
FROM Nombres n, Contenus_mots_cles c, Articles a
WHERE n.id_nombre = c.id_nombre and a.id_article = c.id_article
and (n.id_mot = 139 or n.id_mot = 3431 or n.id_mot = 3991)
and (theme = 'seisme' or theme = 'ouragan' or theme = 'inondation'
or theme = 'tremblement de terre' or theme = 'cyclone'
or theme = 'typhon')
and nombre < 50000
and (nombre NOT BETWEEN 1950 and 2050)
GROUP BY a.theme, year(a.date_article)
ORDER BY year(date_article)
```



*Coût moyen par type de catastrophe naturelle :*

```
SELECT a.theme, AVG(n.nombre) AS Cout
FROM Nombres n, Contenus_mots_cles c, Articles a
WHERE n.id_nombre = c.id_nombre and a.id_article = c.id_article
and (n.id_mot = 4859 or n.id_mot = 2341)
and (theme = 'seisme' or theme = 'ouragan' or theme = 'inondation'
or theme = 'tremblement de terre' or theme = 'cyclone'
or theme = 'typhon')
and nombre < 100000000000
and (nombre NOT BETWEEN 1950 AND 2050)
GROUP BY a.theme;
```

*Coût moyen par année :*

```
SELECT year(a.date_article), AVG(n.nombre) AS Cout
FROM Nombres n, Contenus_mots_cles c, Articles a
WHERE n.id_nombre = c.id_nombre and a.id_article = c.id_article
and (n.id_mot = 4859 or n.id_mot = 2341)
and (theme = 'seisme' or theme = 'ouragan' or theme = 'inondation'
or theme = 'tremblement de terre' or theme = 'cyclone'
or theme = 'typhon')
and nombre < 100000000000
and (nombre NOT BETWEEN 1950 AND 2050)
GROUP BY year(date_article)
ORDER BY year(date_article);
```

*Coût moyen par type de catastrophe naturelle et par année :*

```
SELECT a.theme, year(a.date_article), AVG(n.nombre) AS Cout
FROM Nombres n, Contenus_mots_cles c, Articles a
WHERE n.id_nombre = c.id_nombre and a.id_article = c.id_article
and (n.id_mot = 4859 or n.id_mot = 2341)
and (theme = 'seisme' or theme = 'ouragan' or theme = 'inondation'
or theme = 'tremblement de terre' or theme = 'cyclone'
or theme = 'typhon')
and nombre < 100000000000
and (nombre NOT BETWEEN 1950 AND 2050)
GROUP BY a.theme, year(date_article)
ORDER BY year(date_article);
```

## Annexe 4 : Requêtes pour la carte

*Nombre moyen de morts et nombre d'articles par pays et par thème :*

```
SELECT T1.mot, T2.theme, AVG(T2.Nombre_morts), count(T1.id_article)
FROM (select id_article, mot
from Contenus_mots_cles c2, Mots m2
where c2.id_mot = m2.id_mot and entite = 'PAYS') T1,
(SELECT a.theme, AVG(n.nombre) as Nombre_morts, a.id_article
FROM Nombres n, Contenus_mots_cles c, Articles a
WHERE n.id_nombre = c.id_nombre and a.id_article = c.id_article
and (n.id_mot = 139 or n.id_mot = 3431 or n.id_mot = 3991)
and (theme = 'seisme' or theme = 'ouragan' or theme = 'inondation'
or theme = 'tremblement de terre' or theme = 'cyclone'
or theme = 'typhon')
and nombre < 50000
and (nombre NOT BETWEEN 1950 and 2050)
and a.id_article in (select id_article from Contenus_mots_cles c2, Mots m2 where c2.id_mot
= m2.id_mot and entite = 'PAYS')
GROUP BY a.theme, a.id_article) T2
WHERE T1.id_article = T2.id_article
GROUP BY T2.theme, T1.mot
```

*Nombre moyen de morts et nombre d'article par pays :*

```
SELECT T1.mot, AVG(T2.Nombre_morts), count(T1.id_article)
FROM (select id_article, mot
from Contenus_mots_cles c2, Mots m2
where c2.id_mot = m2.id_mot and entite = 'PAYS') T1,
(SELECT AVG(n.nombre) as Nombre_morts, a.id_article
FROM Nombres n, Contenus_mots_cles c, Articles a
WHERE n.id_nombre = c.id_nombre and a.id_article = c.id_article
and (n.id_mot = 139 or n.id_mot = 3431 or n.id_mot = 3991)
and nombre < 50000
and (nombre NOT BETWEEN 1950 and 2050)
and a.id_article in (select id_article from Contenus_mots_cles c2, Mots m2 where c2.id_mot
= m2.id_mot and entite = 'PAYS')
GROUP BY a.id_article) T2
WHERE T1.id_article = T2.id_article
GROUP BY T1.mot
```

*Nombres moyen de morts par pays pour les années 2000 et 2017*

```
SELECT T1.mot, year(T2.date_article), AVG(T2.Nombre_morts) as Nombre_morts
FROM (select id_article, mot
from Contenus_mots_cles c2, Mots m2
where c2.id_mot = m2.id_mot and entite = 'PAYS') T1,
(SELECT AVG(n.nombre) as Nombre_morts, a.id_article, date_article
FROM Nombres n, Contenus_mots_cles c, Articles a
WHERE n.id_nombre = c.id_nombre and a.id_article = c.id_article
and (n.id_mot = 139 or n.id_mot = 3431 or n.id_mot = 3991)
and nombre < 50000
and (nombre NOT BETWEEN 1950 and 2050)
and a.id_article in (select id_article from Contenus_mots_cles c2, Mots m2 where c2.id_mot
= m2.id_mot and entite = 'PAYS')
GROUP BY a.id_article, date_article) T2
WHERE T1.id_article = T2.id_article
GROUP BY mot, year(date_article)
HAVING year(date_article) = 2000 or year(date_article) = 2017
ORDER BY mot, year(date_article)
```

## Annexe 5 : Carte des pays cités pour les catastrophes naturelles

