

# Représentation des catastrophes naturelles sur un portefeuille : les nouveaux enjeux en cartographie



**Lisa Volte**  
IPE Conseil



**Pierre Mouilhade**  
IPE Conseil



**Nicolas Baradel**  
PGM Solutions

# Qui sommes-nous ?



## **IPE & IPE Conseil** Réassurance & Conseil en Actuariat

Depuis 2017, IPE est un courtier de réassurance indépendant opérant sur les branches non VIE.

IPE Conseil créé en 2019, société de conseil en Actuariat spécialisée en réassurance, risques climatiques, refonte produit et tarification.



## **Lisa Volte** Consultante Réassurance

Lisa, Consultante data scientist développe des modèle CAT et pilote des projets Big Data sur les accumulations de risques non-VIE pour le compte de réassureurs. Experte Cartographie.



## **Pierre Mouilhade** Président

16 ans d'expérience en Banque d'investissement, Assurance et Réassurance.

Expertise en réassurance structurée, modélisation CAT & Actuariat tarification produit.

# Qui sommes-nous ?

**PGM Solutions** Solutions R & Python

► **PGM Solutions** Depuis 2013, développe la solution RPGM qui facilite la création et le déploiement d'applications R et Python par les équipes métiers.



**Nicolas Baradel** Co-fondateur

Docteur en mathématiques appliquées à la finance, expert en développement d'applications avec R et Python. Auteur du livre « Langage R : Applications à la Statistique, à la Finance et à l'Actuariat ».

# Utilité de la cartographie

En assurance non-vie, sur le dommage au bien notamment :

- Les **expositions** sont « localisées »
- Une partie croissante des **facteurs de risques** sont géographiques :
  - Zoniers de risque (Délinquance, Aléa grêle etc...)
  - Composition des sols (Aléa retrait gonflement des sols etc.)
  - Topographie / hydrographie

La cartographie permet donc une **représentation des portefeuilles** d'assurance mais également de superposer des facteurs de risque où évènements spécifiques (tempête, inondation, émeutes etc...).

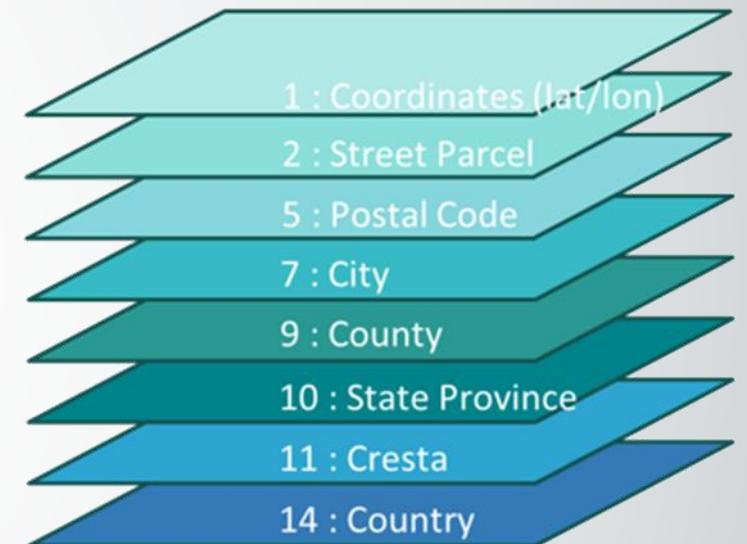
# La qualité des données en Assurance

La variété des produits d'assurance, les multiples créneaux de distribution induisent une **qualité variable de la localisation** des expositions (adresse) :

- Remontée de bordereaux avec des expositions potentiellement agrégées sur des localisations à précision variable
- Police multi sites dont la donnée de chaque site n'est pas renseignée de manière détaillée
- Différentes versions de « tunnel » de souscription sur un même produit
- Adresse mal renseignée par le souscripteur ou l'intermédiaire d'assurance
- Etc...

# La difficulté du géocodage

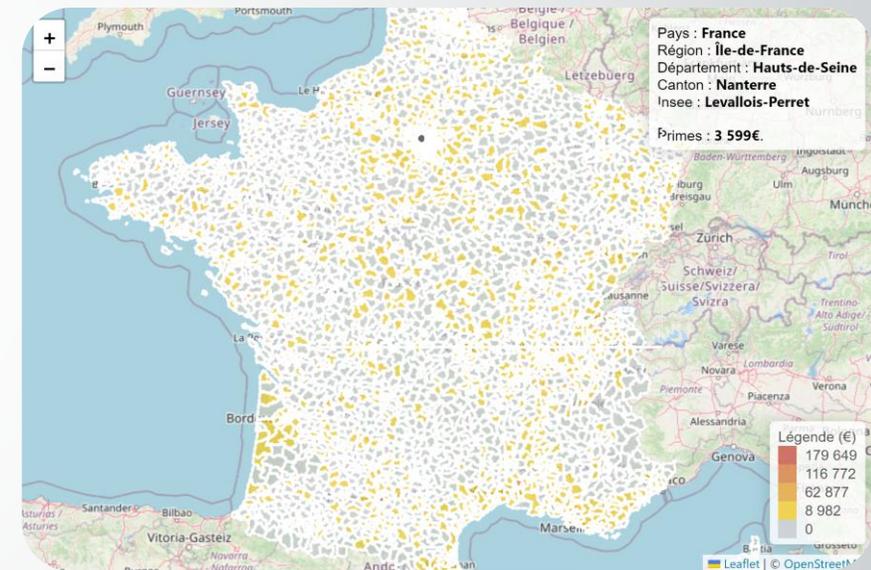
- Le « **Géocodage** » consiste à attribuer un point Latitude/Longitude à une localisation qui peut être une adresse plus ou moins précise.
- Un point Latitude/Longitude n'est donc **pas représentatif d'une localisation exacte**. Par exemple: une exposition localisée à « Paris » serait géocodée au centroïde soit en plein milieu de la Seine.
- A titre illustratif, l'exposition d'un portefeuille peut être « géocodée » avec **14 niveaux de précision différents** dans les outils de modélisation CAT de RMS.
- La solution cartographique doit donc s'adapter à la précision des données



# La difficulté de la représentation

## Lisibilité de la carte

- Une carte a des limites en affichage par marqueur.
- Une carte représentée par des marqueurs seulement est illisible au-delà d'un certain seuil et ne présage pas de l'accumulation du portefeuille.
- A un niveau de zoom correspond une représentation par polygone (ex. région / département / code INSEE). Une représentation par code INSEE est illisible au niveau France.

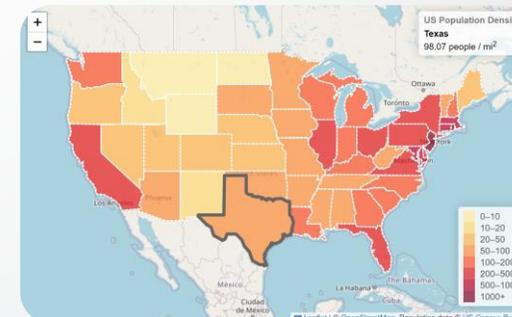
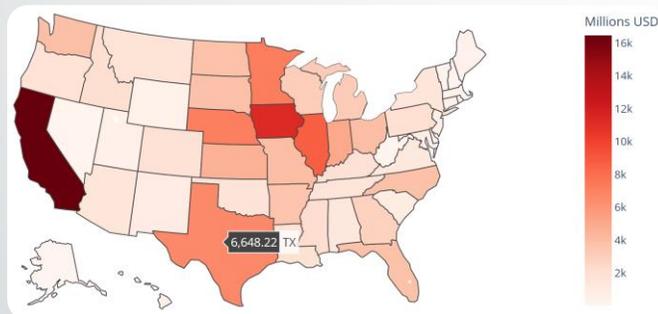


# Usage simple de la cartographie

**Données statiques** Plotly – Leaflet (RPGM ou rshiny ou php)

**Plotly / Leaflet / OpenLayer:** bibliothèque JavaScript de cartes interactives et de visualisations cartographiques dans les applications web, spécialisée pour cela, également utilisée dans Python et R.

1. Chargement de la carte avec **toutes les données**
2. Interaction dans la carte avec les données chargées
3. Pas de possibilité d'envoyer de nouvelles données sans recharger toute la carte



# Limitations techniques

**Données statiques** Plotly – Leaflet (RPGM ou rshiny ou php)

1. Important chargement si beaucoup de données, tout est envoyé à tous les niveaux
2. Ne gère pas les données volumineuses, crash en général avant de recevoir 1 millions de données
3. Pas d'interactivité en dehors de la carte : on peut interagir, mais limitée dans la carte

# Solution « dédiée »

## Une solution toute faite

1. Peut nécessiter d'envoyer les données sur un serveur distant hors UE
2. Coût prohibitif
3. Des limitations techniques demeurent :
  - le fond de carte doit être pré-généré par niveau de résolution.
  - Un portefeuille peut être visualisé selon plusieurs critères de sélection (ex : Ligne de Produit, Type d'Occupation, etc.). Il y a des milliers de combinaisons possibles et chaque combinaison nécessite une carte ad hoc.

# Résoudre le problème

## Mettre un flux temps réel Leaflet/OpenLayer

La limitation précédente provient du fait de devoir envoyer **toutes les données** multi niveau, ce qui n'est pas possible en pratique : **temps de chargement, crash.**

La solution est de construire un environnement qui n'envoie **que les données nécessaires à l'écran**, c'est-à-dire

1. Les données associées au zoom courant sur la carte
2. Uniquement les données qui doivent s'afficher à l'écran

Chaque mouvement dans la carte fera **un appel léger à des données** qui seront chargées rapidement **à la volée.**

# Résoudre le problème



# Solution 1 : Une application web

## Application web

Il s'agit des langages standards des sites et applications **web**, dans lequel Leaflet/OpenLayer est conçu.

Nécessite une application et un **déploiement web** qui ne peut être à la seule main de l'actuaire.

C'est possible et **viable**, mais il faut être conscient de la **difficulté pour un actuaire**.

L'idéal serait d'avoir la même chose dans un langage naturel pour les calculs actuariels et où l'actuaire est chez lui : **R ou Python**.



# Solution 2 : Une app RPGM

**Application RPGM** Technologies R et ou Python.

Permet de gérer les interactions Javascript Leaflet depuis **R ou Python**

Leaflet envoie des informations à R ou Python (nouvel emplacement de l'utilisateur, nouveau niveau de zoom).

Il suffit d'envoyer à Leaflet, **depuis** R ou Python, les données à afficher qui correspondent au niveau de zoom et aux coordonnées actuelles de la carte.

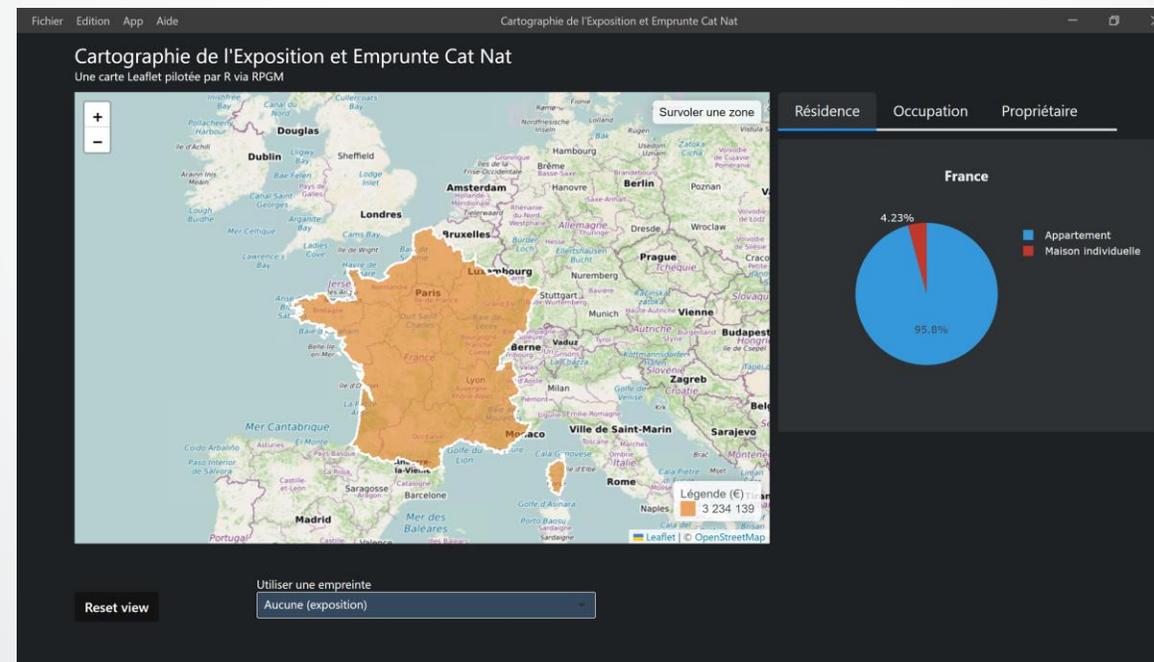
On peut construire **toute l'application** avec R ou Python, l'interface utilisateur se développe par ailleurs **sans code**.

Pas limité à Leaflet (ex : OpenLayer).

# Solution 2 : Une app RPGM

**Démonstration** Une app RPGM avec leaflet.

**Regardons en direct !**



# Conclusion

## Un sujet riche et complexe

Le sujet devient difficile si on veut une exploration **réactive** avec **beaucoup de données**.

Deux manières de l'approcher :

1. Application web complète
2. Application RPGM

La première est **sans licence**, mais **lourde**, en particulier pour un actuaire qui souhaite faire une application complète.

La seconde est avec licence, mais permet d'avoir un **moteur R ou Python** pour piloter la cartographie tout comme le reste de l'application.