

PREMIER RAPPORT

APPRENTISSAGE STATISTIQUE EN ACTUARIAT
ACT-4114

ÉQUIPE 09

Rapport Inondations en Californie

Par

Maryjane BASTILLE
Danny LAROCHELLE
Henri LEBEL
ISABELLE LEGENDRE
Félix-Antoine PARIS

Numéro d'identification

111 268 504
111 174 586
111 286 185
536 768 666
536 776 223

*Travail présenté à
Monsieur*

OLIVIER CÔTÉ

13 MARS 2023



UNIVERSITÉ
LAVAL

Faculté des sciences et de génie
École d'actuariat

Table des Matières

Introduction	2
Sélection des variables	2
Sélection des observations	2
Imputation des données manquantes	3
Création de la nouvelle variable réponse	5
Analyse exploratoire des données	6
Variables non pertinentes	6
Variable <i>condominiumIndicator</i>	7
Variable <i>communityRatingSystemDiscount</i>	8
Variable <i>dateOfLoss</i>	8
Variable <i>elevatedBuildingIndicator</i>	10
Variables <i>latitude</i> et <i>longitudes</i>	10
Variable <i>locationOfContents</i>	12
Variable <i>lowersFlorElevation</i>	14
Variable <i>numberOfFloorsInTheInsuredBuilding</i>	15
Variable <i>occupancyType</i>	16
Variable <i>totalCoverage</i>	16
Variable <i>primaryResidence</i>	18
Conclusion	19
Bibliographie	20
Annexe	21

Introduction

Sélection des variables

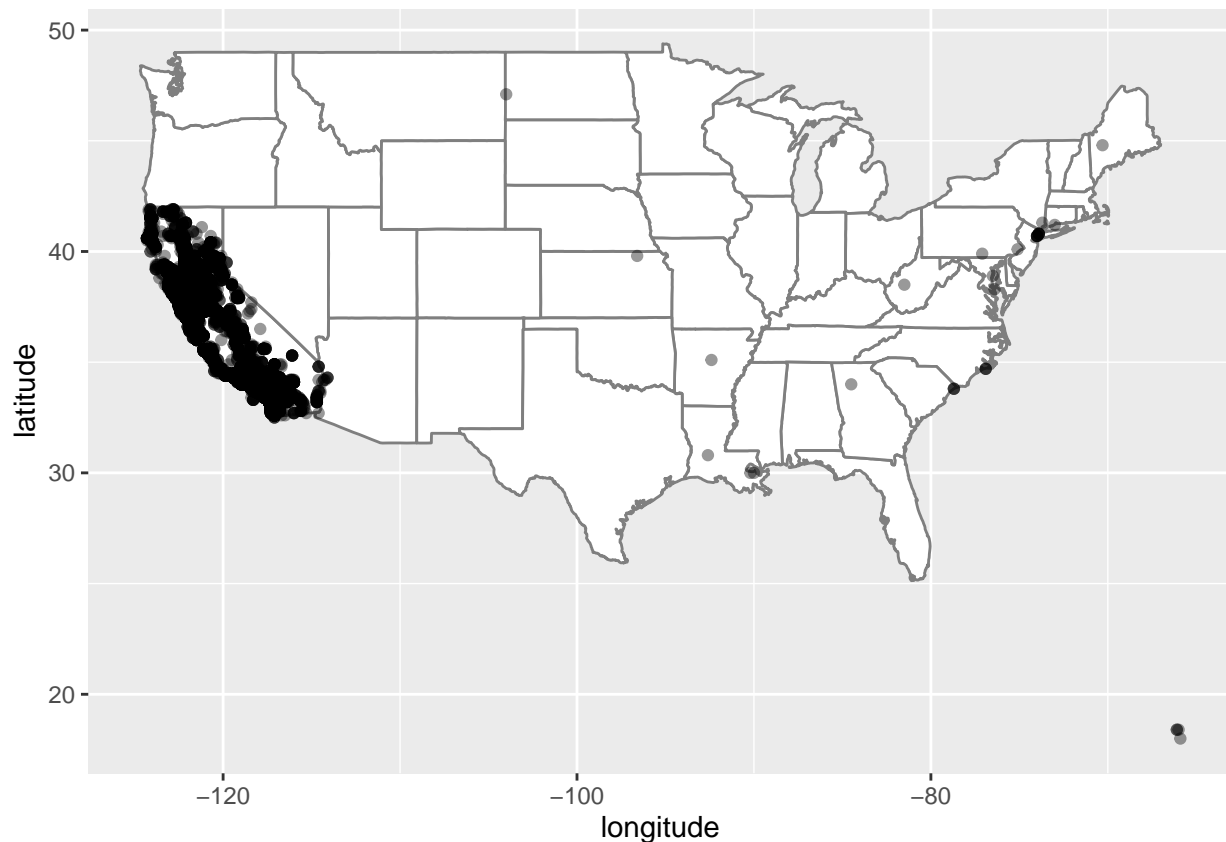
La première étape du travail a consisté à réduire la dimension du jeu de données. En effet, celui-ci est constitué de 41 variables, dont une bonne partie n'étant pas utiles dans le contexte de l'analyse des montants de réclamation.

Sans effectuer aucune analyse statistique, nous avons jugé adéquat de retirer plusieurs variables du modèle, notamment, toutes les variables contenant beaucoup de valeurs manquantes, comme `baseFloodElevation`, `basementEnclosureCrawlSpace`, `elevationCertificateIndicator`, `elevationDifference`, `rateMethod` et `lowestAdjacentGrade`. Ces variables sont aussi toutes issues de l'évaluation de quelques uns des bâtiments assurés, alors que plusieurs autres variables telles que `numberOfFloorsInTheInsuredBuilding`, `originalConstructionDate` ou encore `lowestFloorElevation` auront un impact probablement plus marqué sur le modèle sans devoir nécessiter un travail ardu et approximatif d'estimation d'une grande quantité de données manquantes.

Nous avons aussi pris la décision d'enlever les variables temporelles à l'exception de la date de construction du bâtiment (`originalConstructionDate`) et la date du sinistre (`dateOfLoss`), puisqu'elles sont les seules variables temporelles pertinentes à notre analyse.

Sélection des observations

Les sujets d'intérêts sont, pour se remémorer, les polices d'assurances couvrant le risque d'inondation pour des bâtiments de l'État de la Californie. Il est donc important de s'assurer que les données proviennent uniquement de la Californie. On peut effectuer cette sélection grâce aux données de coordonnées (latitude et longitude) ainsi qu'avec les code de comtés disponibles dans le jeu de données.

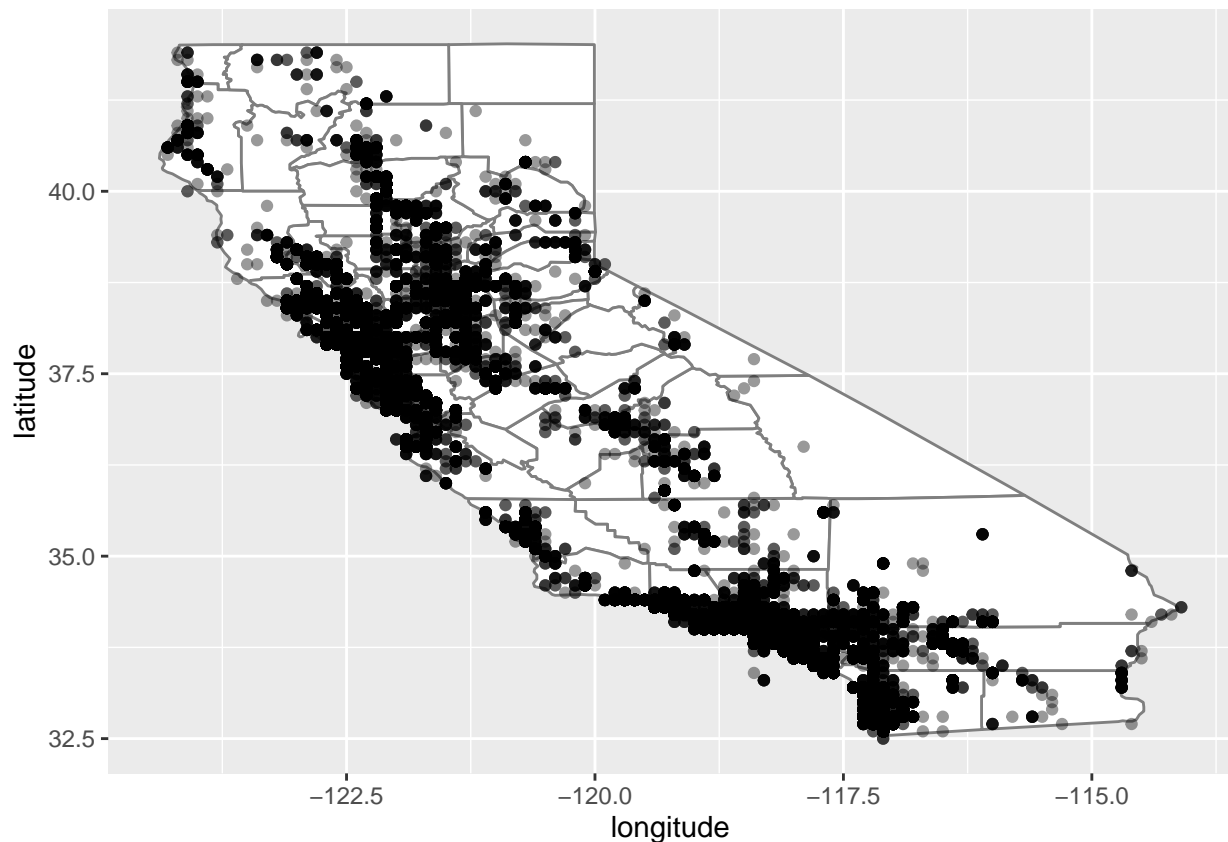


Grâce à cette carte, on peut facilement voir que certaines données ne sont visiblement pas situées en Californie. Ces observations sont retirées du jeu de données.

Ensuite, en observant les codes de comtés des observations restantes, on peut réaliser que trois observations arborent le code "32031", qui appartient au comté de Washoe au Nevada. Ces trois données sont donc retirées du jeu de données.

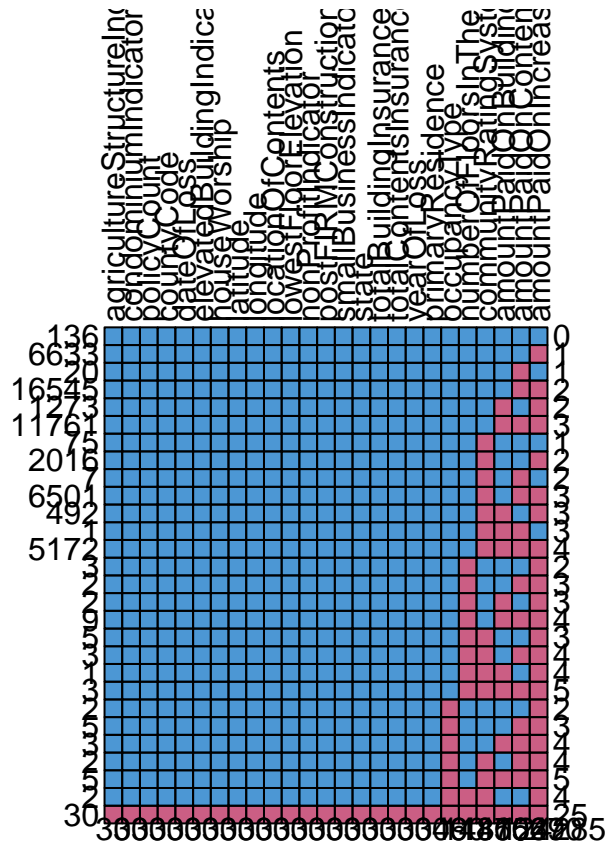
Voici à quoi ressemble la distribution des données restantes sur la carte de la Californie.

```
mapCalifornia <- borders(database = "county", region = "california",
  colour="gray50", fill="white")
ggplot(data = data, aes(x = longitude, y = latitude )) +
  mapCalifornia + geom_point(alpha = .4)
```



Imputation des données manquantes

Le jeu de données comporte plusieurs données manquantes réparties dans multiples variables explicatives. Explorons le patron de non réponse.



Attaquons la variable `communityRatingDiscount` en premier. En effet, celle-ci indique le niveau auquel la police d'assurance a droit à un rabais sur sa prime en fonction de de la zone d'inondation dans laquelle le bâtiment se retrouve. La cote est sur une échelle de 1 à 10, du plus gros rabais pour la classe 1 à l'absence de rabais. La façon la plus intuitive que nous avons pu trouver de gérer les données manquante est de leur attribuer arbitrairement la classe 10, puisque selon l'organisme qui publie ces données, les données manquantes ne participent tout simplement pas au programme de primes.

Ensuite, la variable du nombre d'étages du bâtiment (variable `numberOfFloorsInTheInsuredBuilding`) arborait un certain nombre de données manquantes pouvant être imputées. Nous avons commencé par reclassifier cette la du type d'occupation du bâtiment (`occupancyType`) en trois classes plus intuitives selon les descriptions des 14 classes offertes par le publicateur des données. Une de ces 14 classes n'avait aucune description, nous l'avons attribué à la classe ayant la proportion du nombre d'étages la plus semblable, qui est la classe 2.

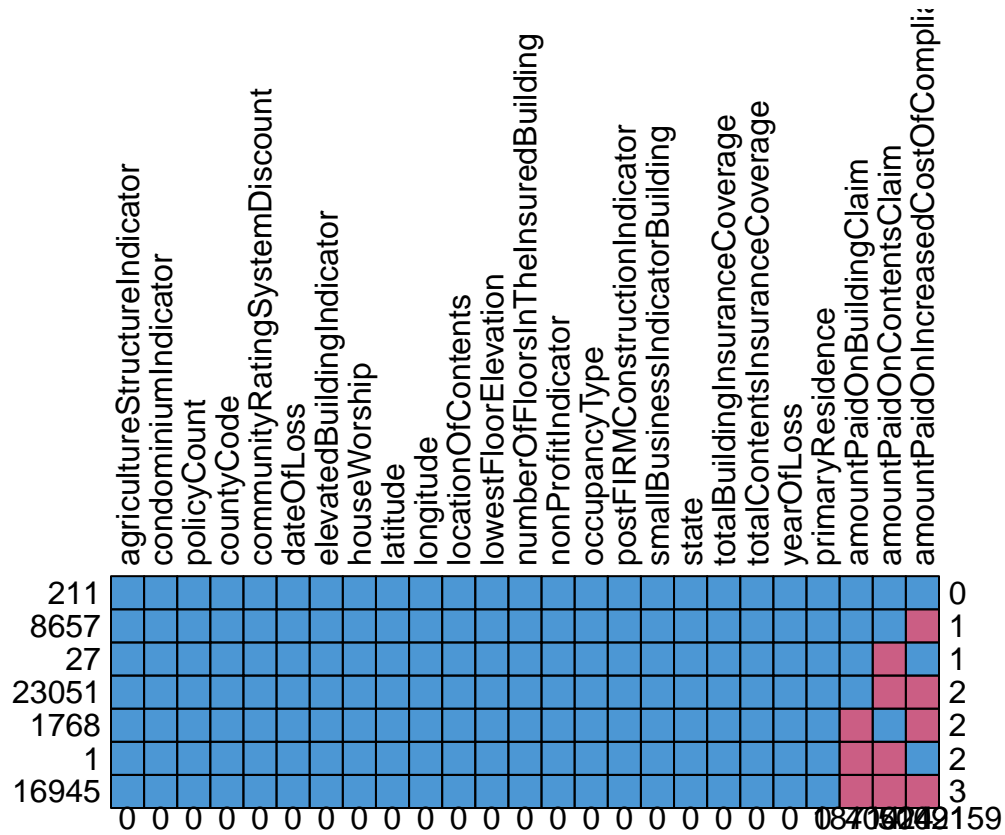
Niveau 1: Résidences familiales Niveau 2: Copropriétés résidentielles Niveau 3: Non-résidentiel

En effet, nous avons identifié la variable du type d'occupation intuitivement comme une variable intimement corrélée avec le nombre d'étages du bâtiment, car elle donne des indices sur la nature du bâtiment. Elle servira donc à l'imputation des données du nombre d'étages.

Nous avons donc imputé les données à l'aide d'un modèle accordant le nombre d'étages aléatoirement en fonction de la distribution du nombre d'étages à l'intérieur d'une même classe d'occupation du bâtiment.

La dernière variable à imputer est l'indicateur du bâtiment étant un condo ou non. On utilise tout simplement notre variable du type d'occupation du bâtiment; si l'observation est dans la catégorie 2, on lui impute une valeur de "1" et une valeur de "0" dans le cas échéant.

Observons maintenant à nouveau notre patron de non réponse.



On remarque que les données manquantes restantes se retrouvent dans les trois variables contenant les montants de réclamation. Nous gèrerons ceci dans la prochaine section.

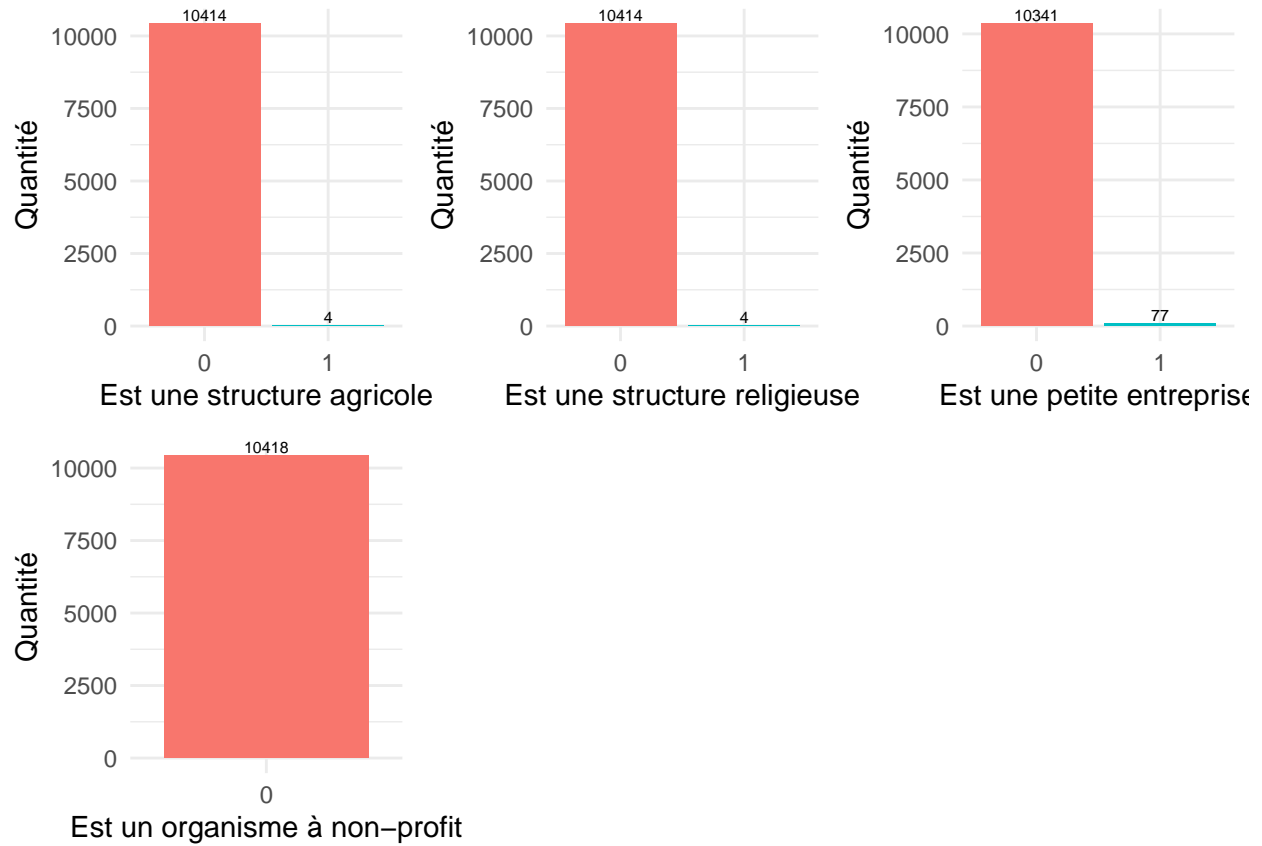
Création de la nouvelle variable réponse

Dans le jeu de données se retrouvent trois colonnes contenant des informations sur les montants de prestations payés en lien avec le bâtiment (`amountPaidOnBuildingClaim`), les biens (`amountPaidOnContentsClaim`) et l'augmentation des coûts en lien avec la conformité (`amountPaidOnIncreasedCostOfComplianceClaim`).

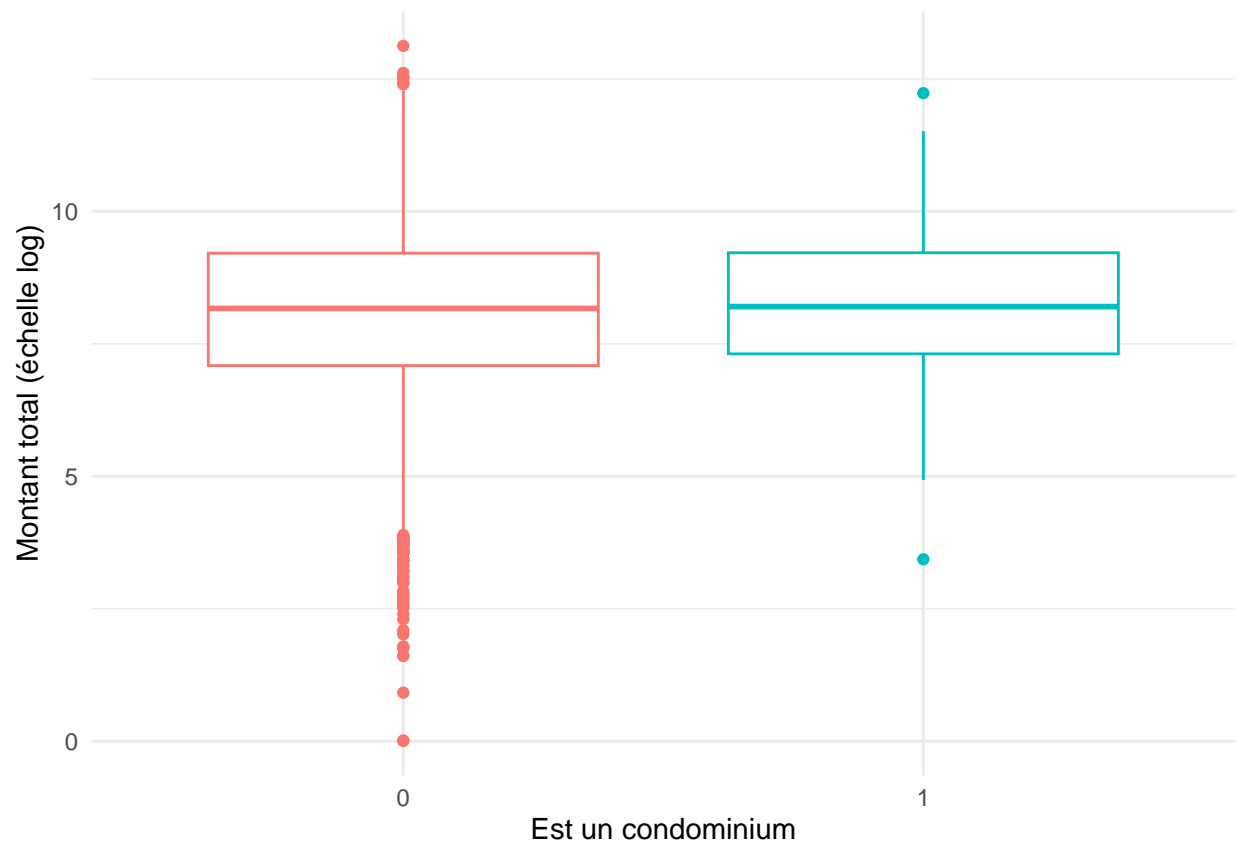
On suppose dans ce cas que les données manquantes peuvent tout simplement se faire attribuer la valeur de 0, indiquant l'absence de paiement dans cette catégorie. Ensuite, nous combinons ces trois variables en créant une nouvelle variable du paiement de prestation total versé au détenteur de police. Celle-ci sera la variable réponse du modèle.

Analyse exploratoire des données

Variables non pertinentes



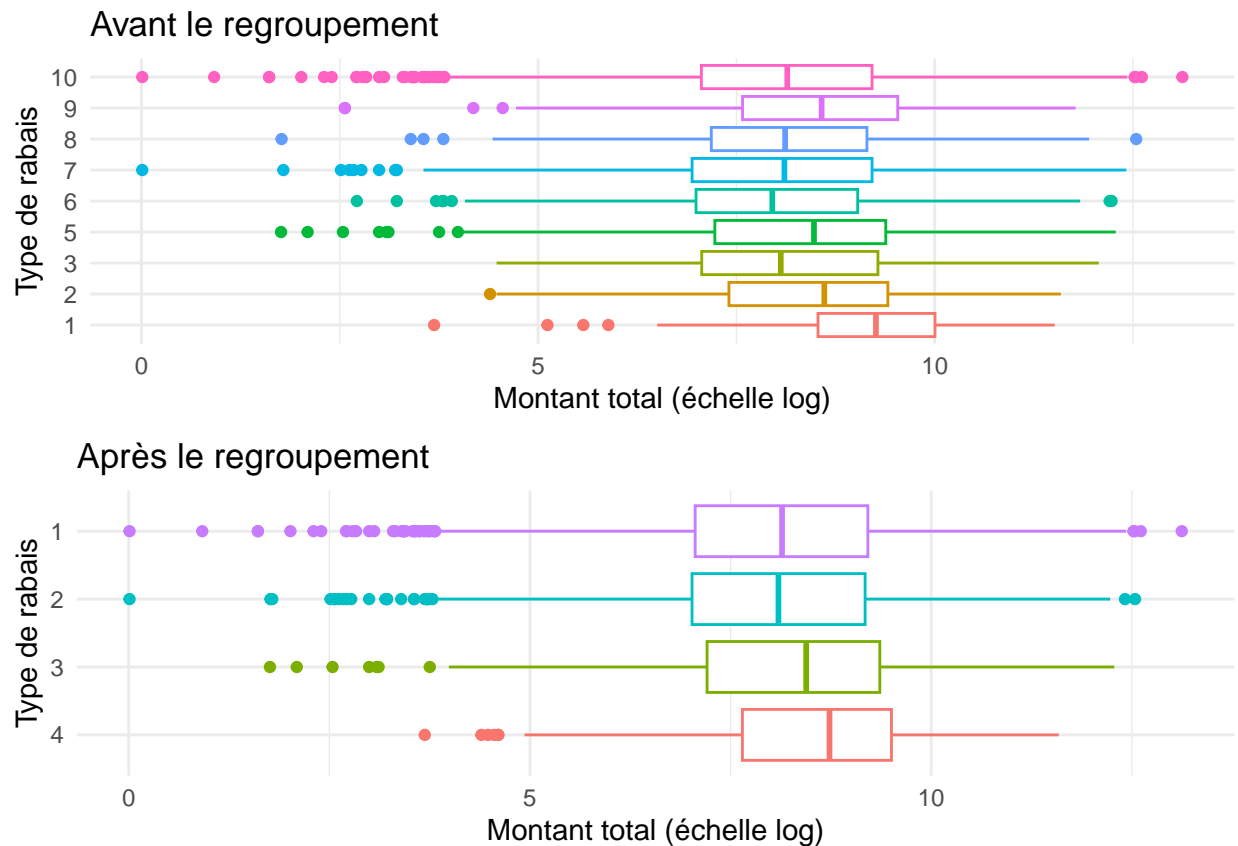
Tel qu'on peut le voir dans les graphiques précédants, les quatres variables indicatives (smallBusinessIndicatorBuilding, agricultureStructureIndicator, houseWorship et nonProfitIndicator) sont trop peu fréquentes pour être significatives, elles seront donc éliminées.

Variable *condominiumIndicator*

	Min	Max	Médiane	Moyenne	Écart-type
Condominium	31.00	204800	3647.65	9993.642	21356.01
Autre	1.01	500000	3524.57	9019.735	18274.78

Variable indicatrice, 1 si l'habitation est un condominium, 0 sinon.

Pour faciliter la visualisation une transformation log est effectuée sur le montant total, cette échelle sera utilisée pour de futurs graphiques.

Variable *communityRatingSystemDiscount*

Cette variable représente le pourcentage de rabais accordé lors de la tarification.

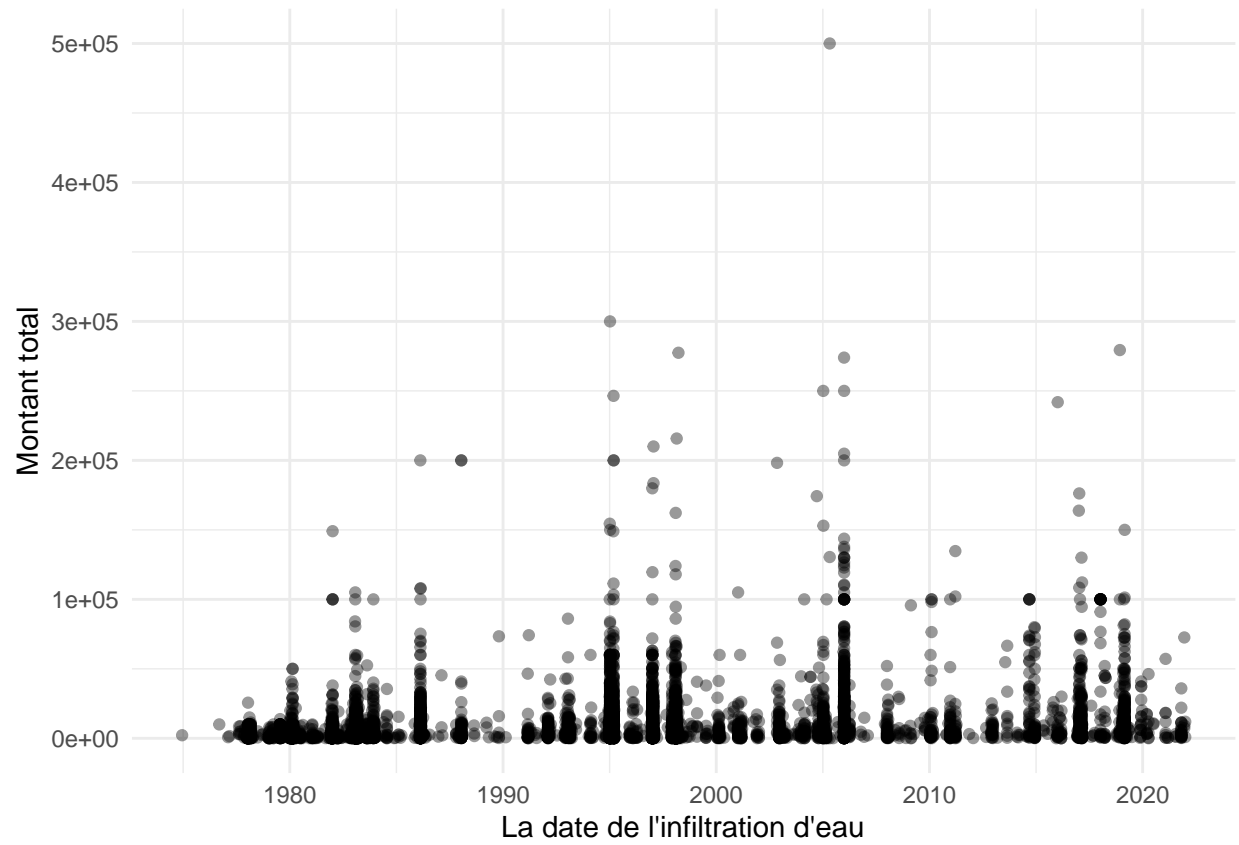
On remarque qu'il y a un trop grand nombre de catégories. Pour cette raison, elles sont réunies en quatre catégories de crédit:

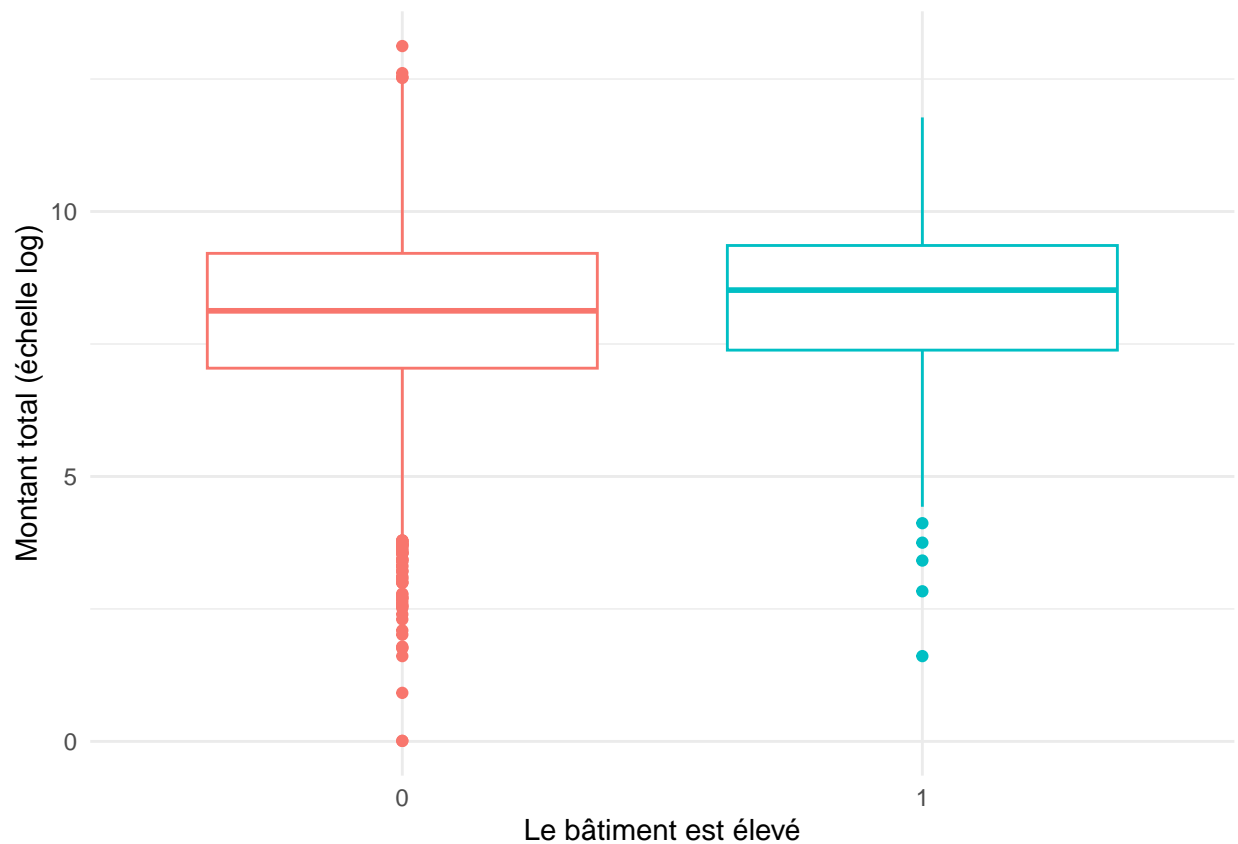
- Niveau 1 : Aucun
- Niveau 2 : 5% - 20%
- Niveau 3 : 25% - 35%
- Niveau 4 : Plus de 40%

Variable *dateOfLoss*

Cette variable temporelle, au jours près, indique la date où s'est produit l'infiltration d'eau dans le bâtiment.

```
ggplot(data, aes(x = dateOfLoss, y = totalAmount))+
  geom_point(alpha = 0.4)+
  theme_minimal()+
  labs(x = "La date de l'infiltration d'eau", y = "Montant total" )
```

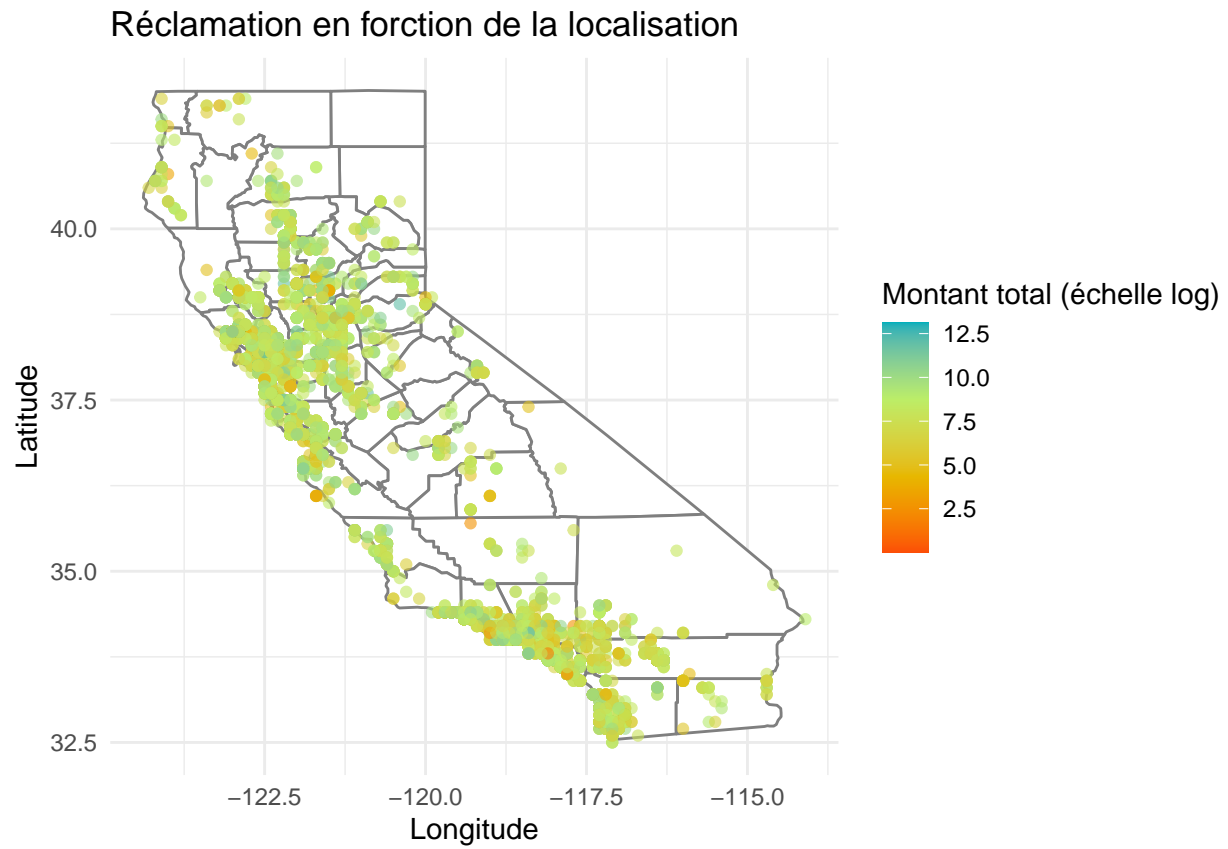


Variable *elevatedBuildingIndicator*

Variable indicatrice, 1 si le bâtiment est élevé, c'est-à-dire, au dessusdu niveau du sol, 0 sinon. Un bâtiment est considéré élevé s'il le plancher le plus bas est au dessus du niveau du sol.

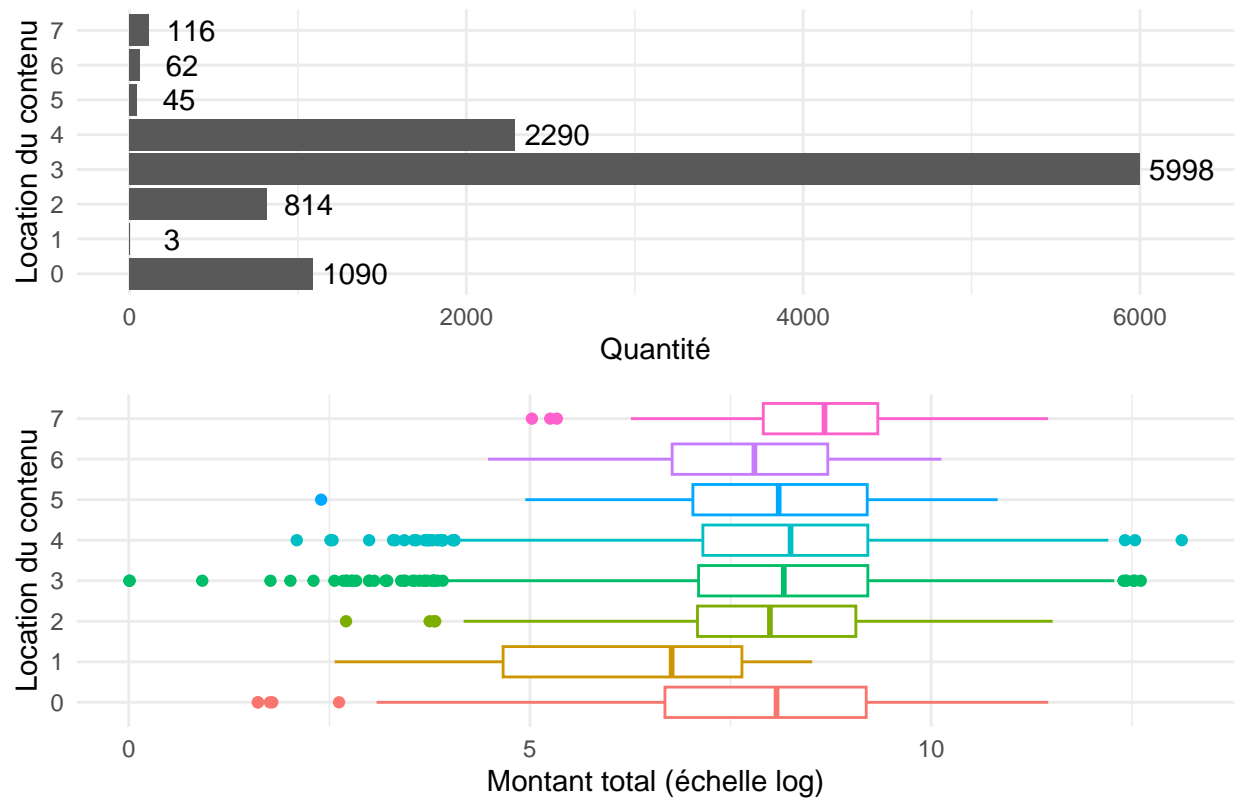
Variables *latitude* et *longitudes*

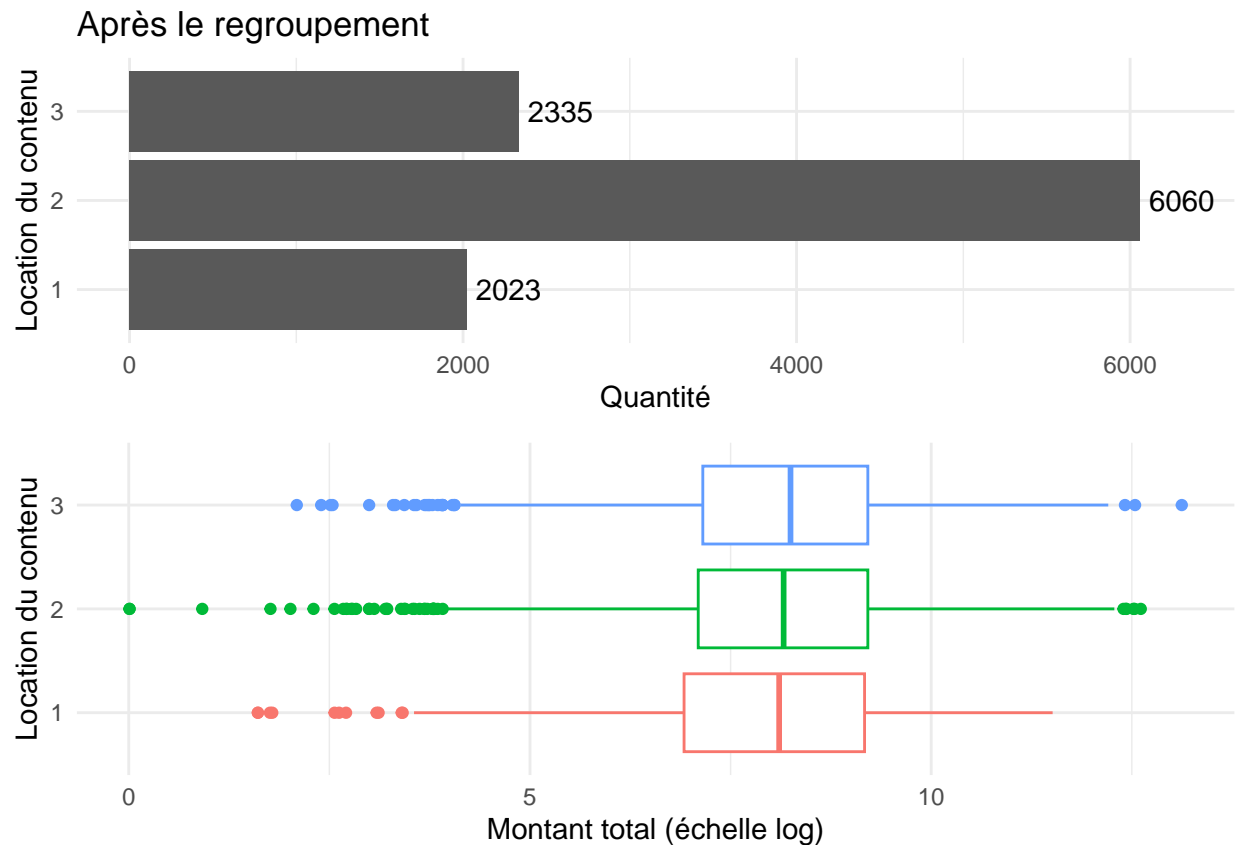
```
mapCalifornia <- borders(database = "county", region = "california",
  colour="gray50", fill="white")
ggplot(data = data, aes(x = longitude, y = latitude, col= log(totalAmount))) +
  mapCalifornia + geom_point(alpha = .6) + scale_color_gradientn(colors = c("#FC4E07", "#E7B800", "darkred")) +
  theme_minimal()+
  labs(x= "Longitude", y="Latitude", title = "Réclamation en forction de la localisation", col ="Montant total")
```



Variable *locationOfContents*

Avant le regroupement

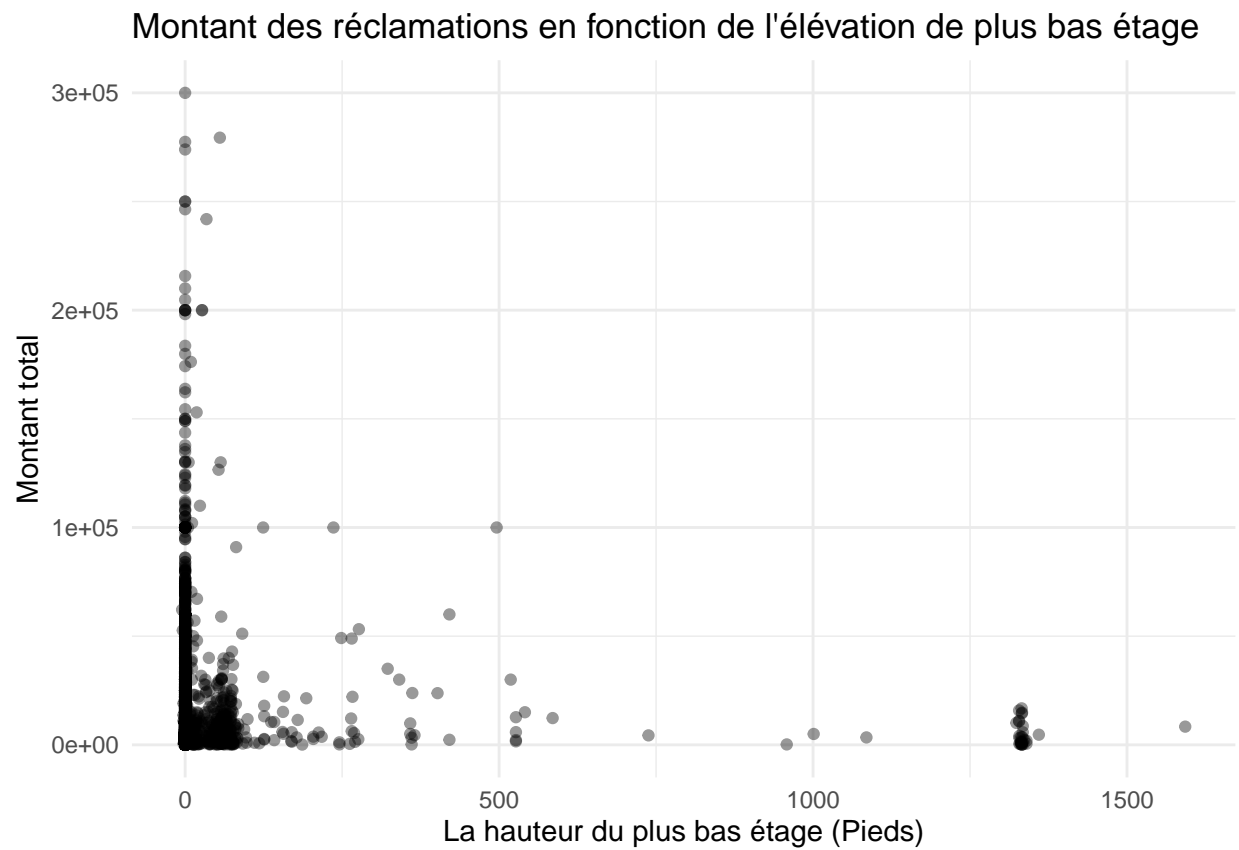




Variable catégorielle indiquant où se trouve le contenu du bâtiment assuré qui a été endommagé.

On remarque pour la variable locationOfContents, qu'il y a plusieurs catégories ne comportant qu'un faible nombre d'observations. Certaines seront donc combinées pour former les nouvelles catégories suivantes:

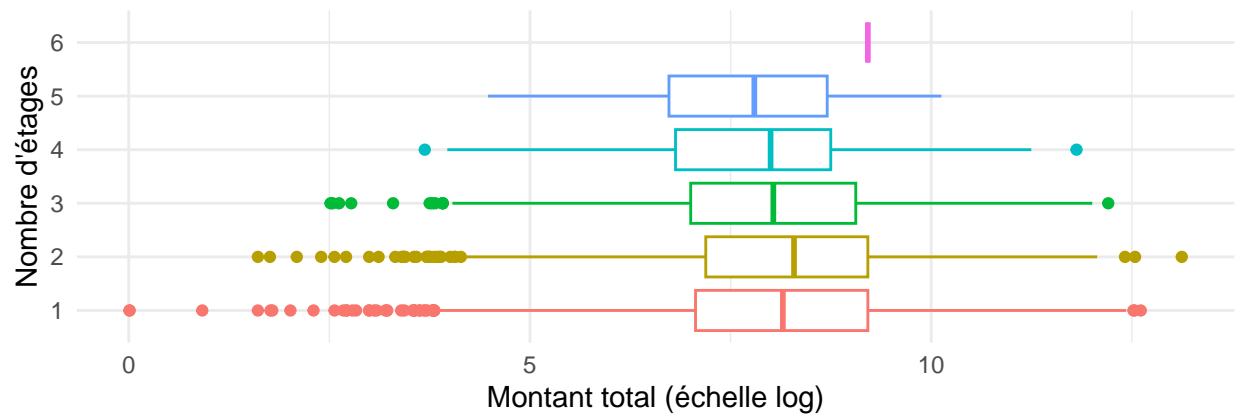
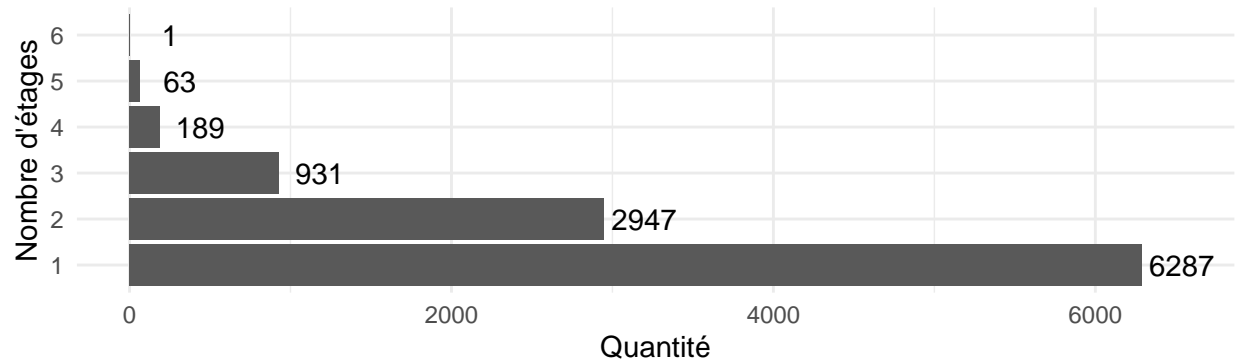
- Niveau 1 : Sous-sol
- Niveau 2 : Premier étage seulement
- Niveau 3 : Premier étage et plus

Variable *lowersFlorElevation*

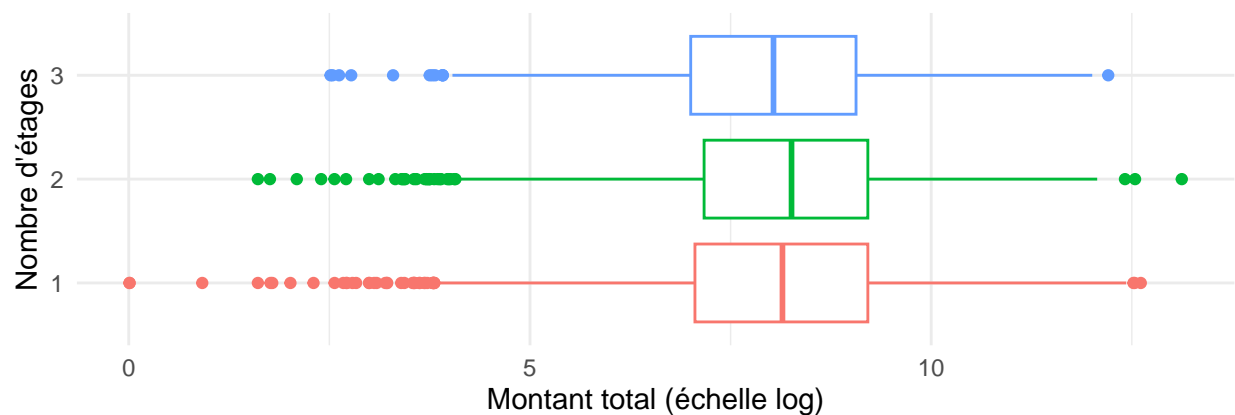
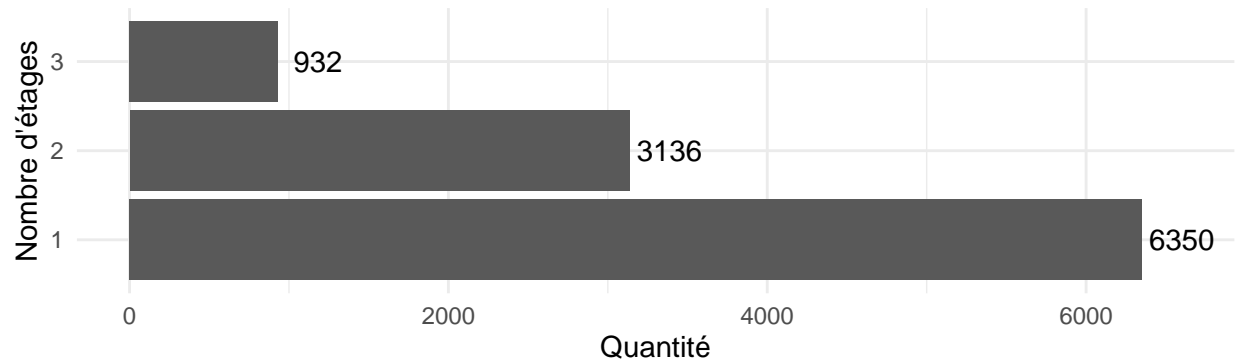
Cette variable représente la hauteur du plus bas étage de l'habitation, en pieds.

Variable *numberOfFloorsInTheInsuredBuilding*

Avant le regroupement



Après le regroupement

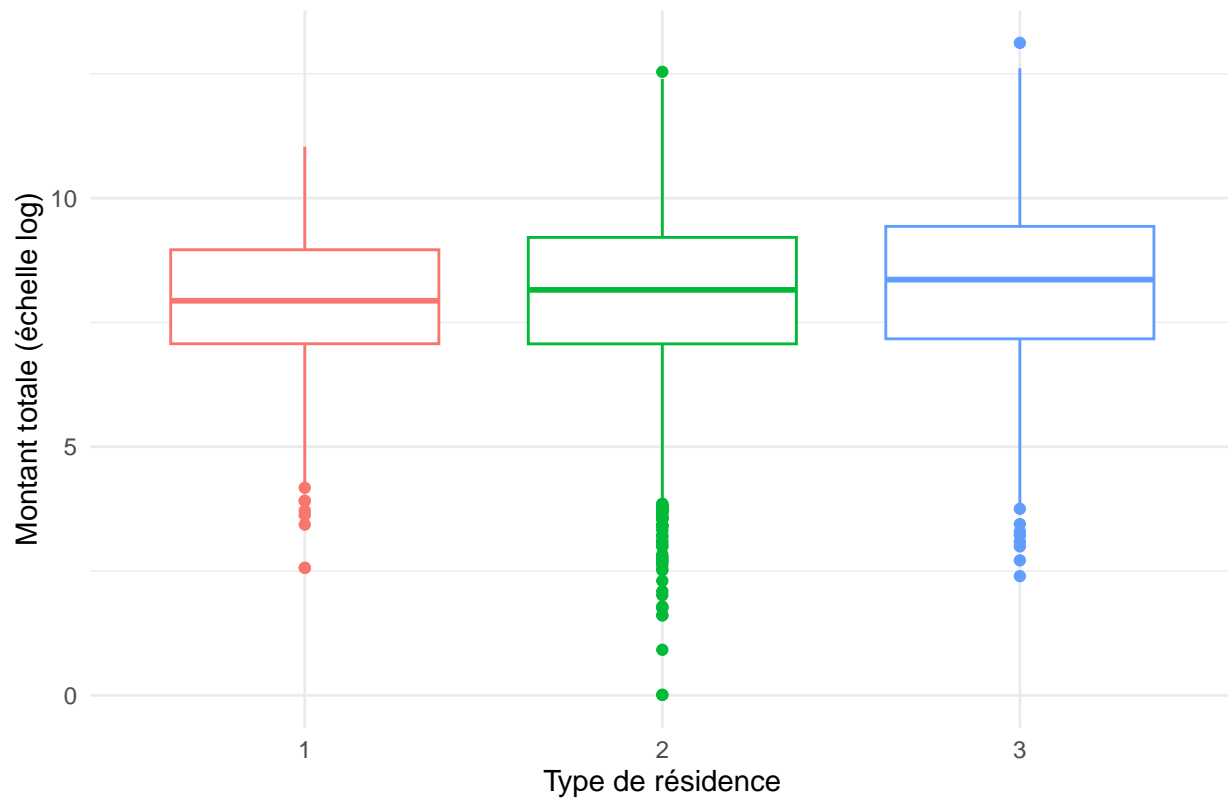


Cette variable comportait 6 catégories d'habitation, nous avons conservé les trois premiers niveaux, et nous avons combinés les trois dernières selon leur nombre d'étages. Nous avons ainsi obtenu les niveaux suivants :

- Niveau 1 : 1 étage
- Niveau 2 : 2 étages
- Niveau 3 : 3 étages et plus

Variable *occupancyType*

Montant des réclamations en fonction du type de résidence



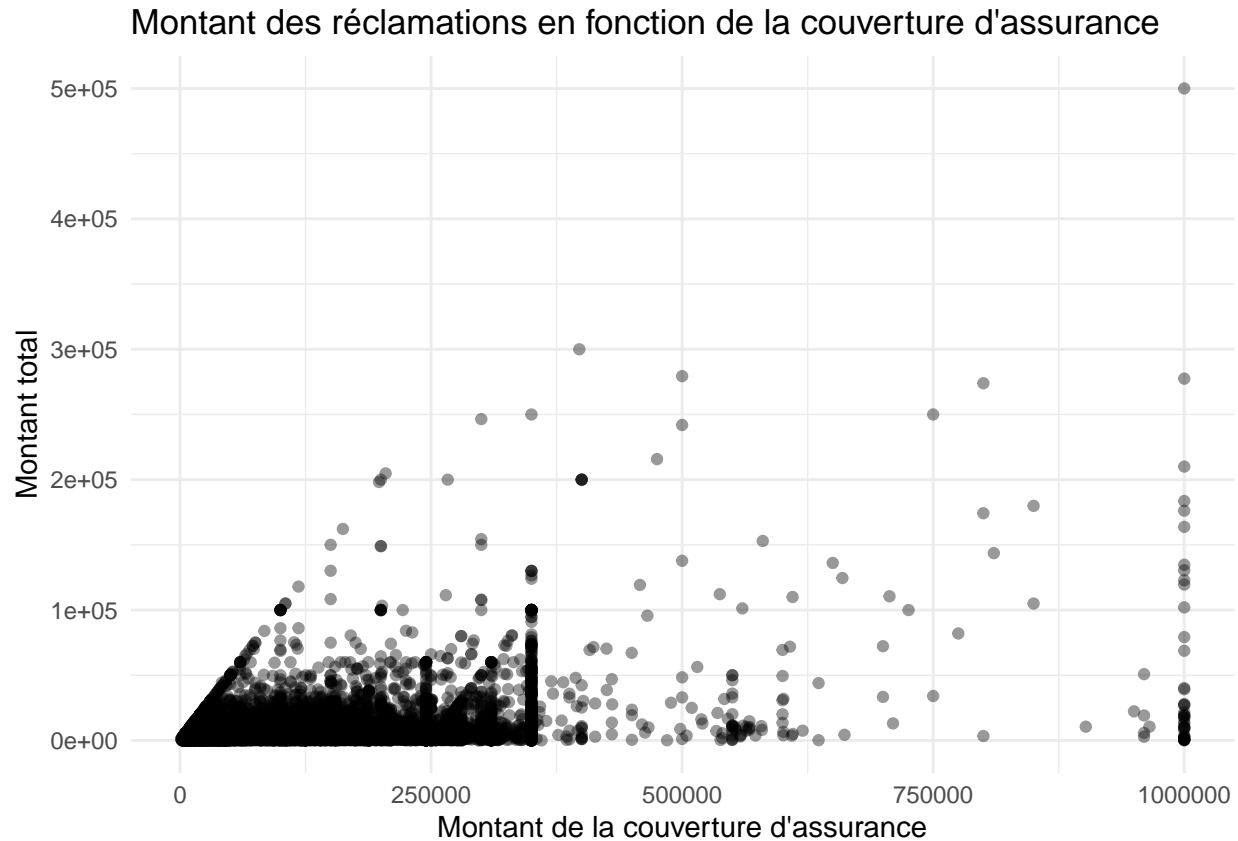
Cette variable catégorielle indique le type de résidence du bâtiment.

- Niveau 1: Résidences familiales
- Niveau 2: Copropriétés résidentielles
- Niveau 3: Non-résidentiel

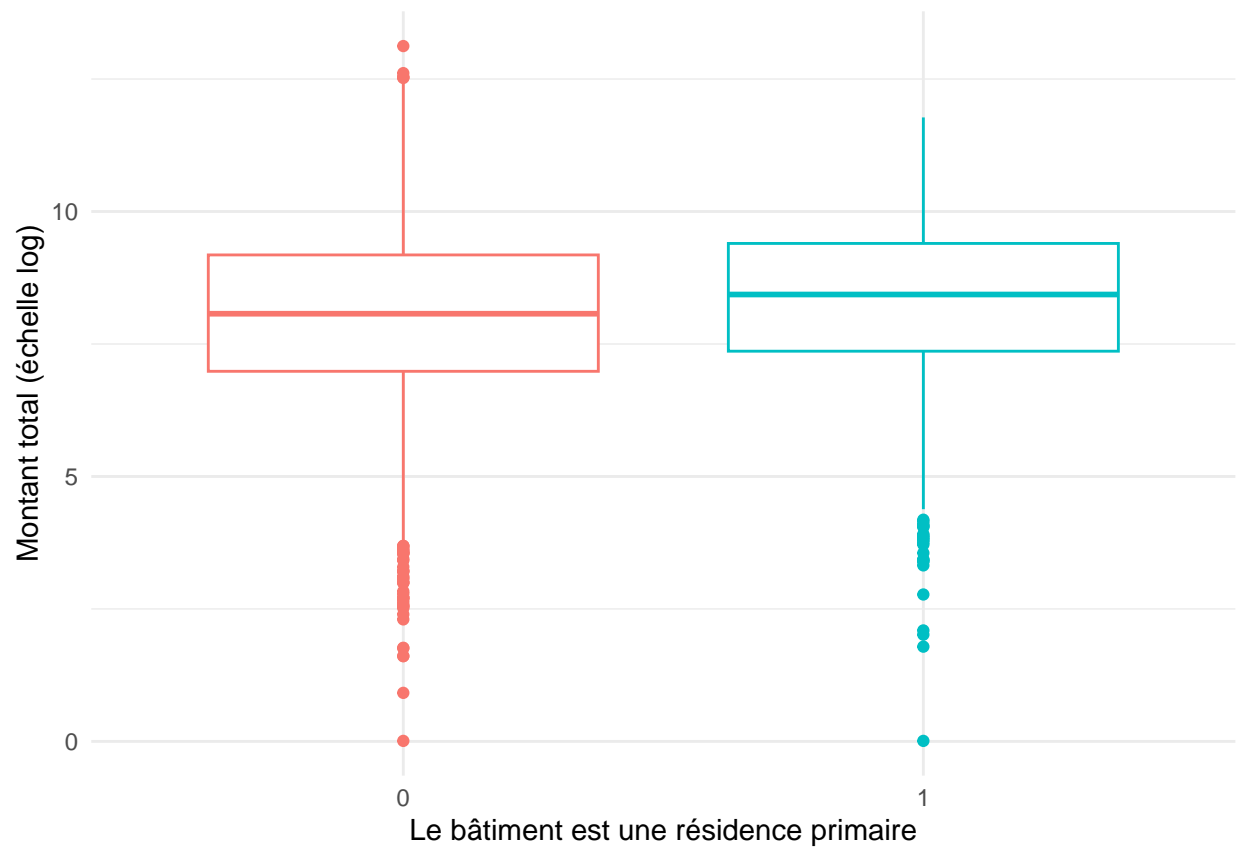
Variable *totalCoverage*

```
data$totalCoverage <- data$totalBuildingInsuranceCoverage + data$totalContentsInsuranceCoverage
```

```
ggplot(data, aes(x = totalCoverage, y = totalAmount))+
  geom_point(alpha = 0.4)+
  theme_minimal()+
  labs(x = "Montant de la couverture d'assurance", y = "Montant total", title = "Montant des réclamations")
```



Dans le jeu de données, les montants de couverture (`totalBuildingInsuranceCoverage` & `totalContentsInsuranceCoverage`) pour le bâtiment et pour les biens personnels étaient séparés, nous avons jugés plus logique de combiner les deux variables en une (`totalCoverage`), puisque nous avons déjà réunis les montants des réclamations.

Variable *primaryResidence*

Variable indicatrice indiquant si le bâtiment assuré est la résidence primaire, 1 si oui, 0 sinon.

Conclusion

Bibliographie

Annexe

Source : FEMA

Lien : <https://www.fema.gov/openfema-data-page/fima-nfip-redacted-claims-v1>

Description : Base de données de réclamations d'assurance faites, par contrat, à la suite d'inondations aux États-Unis. Puisque le jeu de données est trop volumineux, on utilise un subset de l'état de Californie.

Taille du jeu de données : 50 000 observations, 30 variables

Variable réponse : Le montant total payé par réclamation en dollar USD. Pour ce faire, nous devons additionner `amountPaidOnBuildingClaim`, `amountPaidOnContentsCaim` et `amountPaidOnIncreasedCostOfCompliance-Claim`

Exposition: *policyCount* le nombre de polices au contrat

Variables explicatives : 1. *primaryResidence* : Boolean, Y si résidence principale, N sinon 2. *dateOfLoss* : Date, date à laquelle il y a eu l'infiltration d'eau 3. *occupancyType* : Catégorielle, indique l'utilisation et le type du bâtiment 4. *totalBuildingInsuranceCoverage* et *totalContentsInsuranceCoverage* : Numérique, le montant de la couverture au contrat 5. *longitude* et *latitude* : Spatiale, longitude et latitude du bâtiment assuré