

RAPPORT D'EXPLORATION ET DE  
PRÉTRAITEMENT DES DONNÉES D'UN  
PROJET DE MACHINE LEARNING

Accidents routiers en France  
- de 2005 à 2022 -

Emmanuel GAUTIER  
Erika MÉRONVILLE  
Rémi THINEY



# Table des matières

<b>1 Exploration des données</b>	<b>2</b>
1.1 Préambule . . . . .	2
1.2 Chargement des données . . . . .	2
1.3 Découverte des données brutes . . . . .	4
1.3.1 Analyse des données originales : . . . . .	4
1.3.2 Analyse des données brutes concaténées : . . . . .	9
1.3.3 Analyse des données brutes fusionnées : . . . . .	10
1.4 Constats préalables avant le preprocessing des données : . . . . .	17
<b>2 Premier prétraitement des données et son application modélisée</b>	<b>17</b>
2.1 Etape 1 – Restructuration des dataframes : . . . . .	17
2.1.1 Création de fonctions spécifiques . . . . .	17
2.1.2 Considérations préalables . . . . .	20
2.1.3 Lancement du processus . . . . .	20
2.1.4 Contrôles avant fusion . . . . .	21
2.1.5 Fusion des dataframes harmonisés . . . . .	22
2.2 Etape 2 - Prétraitement après fusion des dataframes : . . . . .	23
2.2.1 Contrôles après fusion . . . . .	23
2.2.2 Création de fonctions spécifiques . . . . .	23
2.2.3 Lancement du processus . . . . .	24
2.3 Etape 3 - Modélisation 1 . . . . .	25
2.3.1 Implémentation d'un premier modèle . . . . .	25
2.3.2 Évaluation des performances du modèle . . . . .	27
2.3.3 Analyse des résultats . . . . .	28
<b>3 Deuxième prétraitement des données et ses applications modélisées</b>	<b>28</b>
3.1 Base de structuration des dataframes : . . . . .	29
3.1.1 Structure pour 'caracteristiques' : . . . . .	29
3.1.2 Structure pour 'lieux' : . . . . .	30
3.1.3 Structure pour 'usagers' : . . . . .	31
3.1.4 Structure pour 'vehicules' : . . . . .	31
3.2 Etapes de transformation des dataframes . . . . .	31
3.2.1 Initialisation de l'environnement . . . . .	31
3.2.2 Enregistrement des 4 rubriques de 2019 à 2022 . . . . .	34
3.2.3 Jointure des 4 rubriques . . . . .	34
3.2.4 Suppression d'observations . . . . .	36
3.2.5 Création de variables . . . . .	36
3.2.6 Dichotomisation / catégorisation . . . . .	39
3.2.7 Suppression de variables . . . . .	45
3.2.8 Suppression de doublons . . . . .	47
3.2.9 Équilibrage de la dimension . . . . .	48
3.2.10 Synthèse des actions de préprocessing . . . . .	50
3.2.11 Sauvegarde des données . . . . .	51
3.3 Modélisation 2 . . . . .	51
3.4 Modélisation 3 . . . . .	64

Ce rapport présente le processus d'exploration et de prétraitement d'un jeu de données complexe sur les accidents de la route en France, couvrant la période de 2005 à 2022. L'objectif principal de cette analyse est de préparer les données pour un projet de machine learning visant à modéliser et prédire la gravité des accidents routiers.

De manière générale, les accidents routiers peuvent être déterminés par plusieurs facteurs. En ce sens, les comportements des conducteurs jouent parfois un rôle central : vitesse excessive, alcool, téléphones portables, fatigue sont des acteurs récurrents dans cette tragédie. Les conditions environnementales, telles que la météo et l'état des routes, ajoutent une couche d'incertitude et de danger. Les véhicules eux-mêmes, de par leur âge et leur entretien, influencent le risque d'accident. Les infrastructures routières, avec leurs forces et leurs faiblesses, façonnent également le paysage de cette problématique.

Bien que les statistiques montrent une tendance constante à la baisse des accidents de la route, leur complexité réside dans la multitude de facteurs et de circonstances uniques à chaque incident. Aussi, l'analyse des données est le phare qui éclaire la voie à suivre. En examinant les statistiques des accidents, les profils des conducteurs, et les conditions spécifiques de chaque incident, nous pouvons discerner des patterns, anticiper les risques, et élaborer des stratégies préventives.

Dans ce contexte, notre projet de machine learning a pour objet de recourir aux algorithmes sophistiqués qui permettront de transformer des événements isolés en précieuses données capables de prédire la gravité de futurs accidents. Notre investigation se porte ainsi sur un dataset constitué de plusieurs fichiers téléchargés librement à partir de l'adresse suivante : <https://www.data.gouv.fr/en/datasets/bases-de-donnees-annuelles-des-accidents-corporels-de-la-circulation-routiere-annees-de-2005-a-2022/>.

Le jeu de données, fourni par le gouvernement français, est composé de quatre fichiers principaux :

- caractéristiques : informations générales sur les circonstances de l'accident ;
- lieux : détails sur l'emplacement de l'accident ;
- véhicules : informations sur les véhicules impliqués ;
- usagers : données sur les personnes impliquées dans l'accident.

Ces données sont d'une importance cruciale pour la sécurité routière, car elles peuvent aider à identifier les facteurs de risque et à élaborer des stratégies de prévention plus efficaces.

L'analyse présentée dans ce rapport suit plusieurs étapes clés :

- exploration initiale des données : examen de la structure, des types de variables, et des valeurs pour chaque fichier ;
- analyse détaillée de chaque variable : étude de la distribution, des valeurs manquantes, des outliers, et de l'évolution temporelle ;
- identification des problèmes potentiels : repérage des incohérences, des changements de codification, et des valeurs aberrantes ;
- propositions de prétraitement : suggestions pour le nettoyage, la transformation et la création de nouvelles variables.

Une attention particulière a été portée aux changements survenus dans la collecte et le codage des données au fil des années, notamment les modifications importantes intervenues à partir de 2019.

Ce travail d'exploration et de préparation des données représente une étape décisive dans un processus plus large de data science, allant de la compréhension initiale du problème à la mise en œuvre de solutions basées sur les données. Il est crucial pour assurer la qualité et la fiabilité du modèle de machine learning qui sera développé par la suite, l'objectif final étant de créer un outil capable de prédire la gravité des accidents.

Cette démarche illustre parfaitement comment un Data Scientist contribue à transformer des données brutes en connaissances actionnables et en modèles prédictifs performants, jetant ainsi les bases solides nécessaires pour des analyses avancées et des prises de décision éclairées dans le domaine de la sécurité routière.

# 1 Exploration des données

Les données concernent 72 dataframes au total, soit 1 dataframe par année et par rubrique. Voici les rubriques concernées :

- caractéristiques : qui prend en compte les circonstances générales de l'accident.
- lieux : qui décrit l'endroit de l'accident.
- véhicules : qui énonce les véhicules impliqués dans l'accident.
- usagers : qui relate les usagers impliqués dans l'accident.

Chaque nom de fichier contient le nom de la rubrique, un tiret normal ou bas, l'année avec 4 chiffres et l'extension '.csv'. Les séparateurs sont la virgule, le point-virgule ou la tabulation. Les noms des colonnes sont très courts et principalement écrits en lettres minuscules : ils ont quelques chiffres et majuscules, et suivent les conventions de nommage des variables de python. Avant d'effectuer l'assemblage des dataframes, il est procédé à leur analyse distincte (année par année pour chaque rubrique) en vue d'une meilleure appréhension des données. Plusieurs étapes se succèdent alors :

## 1.1 Préambule

Toutes les bibliothèques nécessaires à la manipulation des données, l'analyse statistique, la visualisation des données, et la préparation des données sont préalablement chargées.

Code Python

```
1 # Importation des librairies
2 import pandas as pd
3 import numpy as np
4 import seaborn as sns
5 import matplotlib.pyplot as plt
6 import logging
7 import math
8 import csv
9 import os
10 import re
11
12 from IPython.display import display
13 from sklearn.preprocessing import StandardScaler
14 from scipy.stats import chi2_contingency
15 from scipy.stats import f_oneway
16 from scipy.stats import spearmanr
17 from tabulate import tabulate
18 from scipy import stats
19 from sklearn.preprocessing import LabelEncoder
```

En configurant un système de journalisation dès le début, nous assurons que toutes les actions, messages d'information, erreurs, et avertissements sont correctement capturés et enregistrés, ce qui facilite la maintenance, le débogage et la documentation du processus.

Code Python

```
1 # Configuration de la journalisation
2 logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s -
    %(message)s')
```

## 1.2 Chargement des données

Les fichiers CSV sur lesquels nous avons travaillé ont comme particularités des variations de format, d'encodage, de nommage ou autres :

- encodage : « utf-8 », « latin1 », « ISO-8859-1 »
  - séparateur : « , », « ; », « \t »
  - connecteur : « - », « \_\_ »
  - nommage : « caractéristiques », « carctéristiques »
  - année : de « 2005 » à « 2022 »

Il nous a donc fallu prendre en compte ces différenciations pour parvenir à une uniformisation de ces derniers. Le code que nous avons créé pour automatiser le processus de chargement des fichiers prend notamment en considération les différents points suivants :

- Détection automatique du délimiteur : La fonction `get_delimiter` analyse un échantillon du fichier CSV pour identifier automatiquement le caractère utilisé comme séparateur (virgule, point-virgule, etc.).
  - Lecture flexible des fichiers CSV : La fonction `read_csv_file` tente de lire le fichier CSV en utilisant différents encodages (UTF-8, Latin-1, ISO-8859-1) pour gérer les variations possibles dans l'encodage des caractères. Elle utilise également le délimiteur détecté précédemment.
  - Chargement systématique des datasets : La fonction `load_datasets` parcourt une série de fichiers CSV basés sur des préfixes spécifiques (caractéristiques, lieux, usagers, véhicules) et des années (de 2005 à 2022). Elle gère les différences de nommage des fichiers selon l'année (utilisation de '\_' ou '-' comme connecteur).
  - Gestion des erreurs et logging : Le code intègre un système de gestion des erreurs et de logging pour suivre le processus de chargement, signaler les problèmes rencontrés (fichiers manquants, erreurs de lecture) et fournir un résumé des datasets chargés avec succès.
  - Flexibilité et réutilisabilité : Le code est conçu pour être facilement adaptable à différents ensembles de données en permettant la spécification des préfixes de fichiers et des années à traiter.

Code Python

```
1 # Détection du délimiteur utilisé dans chaque fichier CSV
2 def get_delimiter(file_path, bytes=4096):
3     try:
4         with open(file_path, 'r') as file:
5             data = file.read(bytes)
6             sniffer = csv.Sniffer()
7             delimiter = sniffer.sniff(data).delimiter
8             return delimiter
9     except Exception as e:
10         logging.error(f"Erreur lors de la détection du délimiteur: {e}")
11         return None
12
13 # Lecture des fichiers CSV avec différents encodages
14 def read_csv_file(file_path):
15     if not os.path.exists(file_path):
16         return None, False, f"Fichier non trouvé: {file_path}"
17
18     delimiter = get_delimiter(file_path)
19     if not delimiter:
20         return None, False, f"Impossible de détecter le délimiteur pour le fichier:
21         {file_path}"
22
23     encodings = ['utf-8', 'latin1', 'ISO-8859-1']
24     for encoding in encodings:
25         try:
26             df = pd.read_csv(file_path, low_memory=False, encoding=encoding,
27                             delimiter=delimiter)
28             return df, True, None
```

```

27     except (UnicodeDecodeError, pd.errors.ParserError) as e:
28         logging.warning(f"Erreur avec l'encodage {encoding} pour le fichier
29                         {file_path}: {e}")
30
31     return None, False, f"Impossible de lire le fichier {file_path} avec les encodages:
32                         {encodings}."
33
34 # Chargement des datasets en fonction des préfixes et des années spécifiés
35 def load_datasets(prefixes, years, base_path='data/raw'):
36     dataframes = []
37
38     for prefix in prefixes:
39         datasets = []
40         for year in years:
41             connector = '_' if year <= 2016 else '-'
42             file_name = os.path.join(base_path, f'{prefix}{connector}{year}.csv')
43             df, success, error = read_csv_file(file_name)
44             if success:
45                 datasets.append({file_name: df})
46             else:
47                 logging.error(error)
48         dataframes.append(datasets)
49
50 # Spécification des années et des préfixes pour les fichiers CSV
51 years = list(range(2005, 2023))
52 prefixes = ['caracteristiques', 'lieux', 'usagers', 'vehicules']
53
54 # Chargement des datasets
55 dataframes = load_datasets(prefixes, years)
56
57 # Log des datasets chargés
58 for prefix, df_list in zip(prefixes, dataframes):
59     logging.info(f'{prefix}: {len(df_list)} datasets chargés.')
60 logging.info(f'Total datasets chargés: {sum(len(dfs) for dfs in dataframes)}.')

```

## Observations :

La prise en compte des caractéristiques propres à chaque dataframe permet la lecture et le chargement de tous les dataframes téléchargés.

### 1.3 Découverte des données brutes

#### 1.3.1 Analyse des données originales :

- **Exploration primaire :**

Chacun des dataframes fait l'objet d'une première analyse isolément, afin de repérer d'éventuelles anomalies. La méthode consiste alors à :

- créer une fonction d'extraction du nom du fichier CSV pour obtenir l'année concernée en utilisant une expression régulière (fonction extract\_year).
- créer une fonction visant à analyser chaque colonne dans tous les datasets (analyze\_column), en collectant les informations suivantes :
  - l'année du dataset,
  - le type de données de la colonne,
  - la valeur la plus fréquente (mode),
  - la proportion de valeurs nulles dans le fichier,
  - la proportion de valeurs nulles par rapport à l'ensemble des données.

- analyser globalement toutes les colonnes de chacun des datasets (fonction analyze\_all\_columns) en :
  - identifiant toutes les colonnes uniques,
  - calculant le nombre total de lignes dans tous les datasets,
  - générant un logging permettant de suivre le processus de chargement de chaque fichier,
  - appellant la fonction analyze\_column pour chaque colonne unique.
- présenter les résultats de l'analyse sous forme de tableau pour chaque colonne de tous les datasets, pour faciliter la comparaison entre les différentes années.

#### Code Python

```

1 # Extraction de l'année du nom de fichier CSV
2 def extract_year(file_name):
3     match = re.search(r'(\d{4})\.csv$', file_name)
4     return match.group(1) if match else None
5
6 # Analyse de chaque colonne des datasets
7 def analyze_column(column, datasets, total_rows):
8     column_results = []
9
10    for dataset in datasets:
11        for file_name, df in dataset.items():
12            if column in df.columns:
13                col_type = df[column].dtype
14                col_mode = df[column].mode()[0] if not df[column].mode().empty else "N/A"
15                null_proportion_file = df[column].isnull().mean() * 100
16                null_proportion_total = df[column].isnull().sum() / total_rows
17                column_results.append([
18                    extract_year(file_name),
19                    col_type,
20                    col_mode,
21                    null_proportion_file,
22                    null_proportion_total
23                ])
24
25    if column_results:
26        print(f"\nColonne: {column} / lignes: {total_rows}")
27        print(tabulate(column_results, headers=[
28            "Année", "Type", "Mode",
29            "Proportion valeurs nulles (fichier)",
30            "Proportion valeurs nulles (total)"
31        ]))
32
33 # Analyse de toutes les colonnes des datasets
34 def analyze_all_columns(datasets):
35     all_columns = set()
36     for dataset in datasets:
37         for file_name, df in dataset.items():
38             logging.info(f'Chargement de {file_name}.')
39             all_columns.update(df.columns)
40
41     total_rows = sum(df.shape[0] for dataset in datasets for file_name, df in
42                      dataset.items())
43
44     # Traitement pour chaque colonne
45     for column in all_columns:
46         analyze_column(column, datasets, total_rows)
47
48 # Analyse des colonnes pour chaque ensemble de datasets
49 for datasets in dataframes:
50     analyze_all_columns(datasets)

```

## Observations :

L'analyse ainsi effectuée a permis de relever les points suivants :

- La variable ‘Num\_Acc’ (ou son équivalent ‘Accident\_Id’) est présente dans chaque dataframe, elle permet de lier les dataframes des différentes rubriques entre eux.
- D'une année à l'autre, le type des variables varie parfois, ce qui doit être pris en compte par la suite pour le prétraitement des données.
- Il existe quelques variables ayant des valeurs nulles, et parfois ces valeurs sont encore plus présentes pour les années antérieures à 2019.
- Certaines variables (comme ‘vma’) ne sont existantes que depuis 2019, alors que d'autres (comme ‘secu’) ont disparu à partir de 2019 lorsqu'elles ne sont pas remplacées par une autre (exemple : ‘Num\_Acc’ qui devient ‘Accident\_Id’ en 2022).

### • Visualisation graphique :

Pour une analyse plus approfondie de la structure et de l'évolution des données, nous avons fait appel aux outils de visualisation. L'objectif était de :

- comprendre la distribution des valeurs numériques et noter la présence d'éventuelles valeurs aberrantes (boxplots) ;
- visualiser la répartition des modalités pour les variables catégorielles (barplots empilés).
- suivre l'évolution de la distribution des modalités par année (lineplots).

Dans ce cadre, notre démarche a consisté à :

- créer des boxplots (create\_boxplot\_grid) qui :
  - génèrent une grille de boxplots pour visualiser la distribution des valeurs numériques d'une colonne spécifique à travers les années,
  - ignorent les colonnes non numériques,
  - permettent une comparaison visuelle rapide ;
- créer des barplots empilés (create\_total\_stacked\_barplot) qui :
  - visualisent la distribution des modalités (valeurs uniques) d'une colonne à travers les années,
  - limitent le nombre de modalités affichées pour éviter la surcharge visuelle,
  - présentent les données sous forme de barres empilées, en montrant l'évolution de la proportion de chaque modalité au fil du temps ;
- créer des lineplots (create\_lineplot\_evolution) qui :
  - montrent l'évolution de la distribution des modalités d'une colonne au fil des années,
  - affichent chaque modalité comme une ligne distincte, permettant de suivre son évolution en pourcentage,
  - limitent également le nombre de modalités pour maintenir la lisibilité ;
- orchestrer les visualisations (one\_column\_generate\_plots) :
  - en combinant les trois types de visualisation (boxplot, barplot, lineplot) pour une colonne donnée,
- analyser globalement (all\_columns\_generate\_plots) :
  - en générant toutes les visualisations graphiques de chaque colonne unique trouvée dans l'ensemble des datasets,

### Code Python

```
1 # Création d'une grille de boxplot
2 def create_boxplot_grid(column, datasets):
3     if not any(pd.api.types.is_numeric_dtype(df[column]) for dataset in datasets for
4         file_name, df in dataset.items() if column in df.columns):
        logging.info(f"La colonne {column} n'est pas numérique.")
```

```

5         return
6
7     n_datasets = len(datasets)
8     n_cols = 5
9     n_rows = math.ceil(n_datasets / n_cols)
10
11    fig, axes = plt.subplots(nrows=n_rows, ncols=n_cols, figsize=(16, n_rows * 2),
12                             squeeze=False)
13    fig.suptitle(f'Boxplots for column: {column}', fontsize=16)
14
15    for ax, dataset in zip(axes.flatten(), datasets):
16        for file_name, df in dataset.items():
17            if column in df.columns and pd.api.types.is_numeric_dtype(df[column]):
18                ax.boxplot(df[column].dropna(), vert=True)
19                ax.set_title(extract_year(file_name))
20                ax.set_xlabel(column)
21                ax.set_ylabel('Values')
22
23    # Masquer les axes non utilisés
24    for ax in axes.flatten()[n_datasets:]:
25        ax.axis('off')
26
27    plt.tight_layout(rect=[0, 0, 1, 0.96])
28    plt.show()
29
30 # Création d'un barplot pour visualiser la distribution des modalités
31 def create_total_stacked_barplot(column, datasets, max_modalities=50):
32     modality_counts = {}
33
34     for dataset in datasets:
35         for file_name, df in dataset.items():
36             if column in df.columns:
37                 modality_count = df[column].value_counts()
38                 modality_counts[file_name] = modality_count
39
40     all_modalities = set()
41     for counts in modality_counts.values():
42         all_modalities.update(counts.index)
43
44     if len(all_modalities) > max_modalities:
45         logging.warning(f"Le nombre de modalités uniques dans la colonne {column} excède
46                         le seuil de {max_modalities}. Aucun bar plot généré.")
47     return
48
49     modality_data = {modality: [] for modality in all_modalities}
50     years = [extract_year(file_name) for file_name in modality_counts.keys()]
51
52     for modality in all_modalities:
53         for file_name in modality_counts.keys():
54             count = modality_counts[file_name].get(modality, 0)
55             modality_data[modality].append(count)
56
56     df_modalities = pd.DataFrame(modality_data, index=years).transpose()
57
58     df_modalities.plot(kind='bar', stacked=True, figsize=(15, 7), colormap='viridis')
59     plt.title(f'{column}')
60     plt.xlabel('Modalities')
61     plt.ylabel('Count')
62     plt.legend(title='Years', bbox_to_anchor=(1.05, 1), loc='upper left')
63     plt.tight_layout()
64     plt.show()

```

```

64
65 # Création d'un lineplot pour visualiser l'évolution de la distribution des colonnes
66 def create_lineplot_evolution(columnn, datasets, max_modalities=12):
67     modality_counts = {}
68
69     for dataset in datasets:
70         for file_name, df in dataset.items():
71             if columnn in df.columns:
72                 year = extract_year(file_name)
73                 modality_count = df[columnn].value_counts(normalize=True) * 100
74                 if year not in modality_counts:
75                     modality_counts[year] = modality_count
76                 else:
77                     modality_counts[year] = modality_counts[year].add(modality_count,
78                                         fill_value=0)
78
79     all_modalities = set()
80     for counts in modality_counts.values():
81         all_modalities.update(counts.index)
82
83     if len(all_modalities) > max_modalities:
84         logging.warning(f"Le nombre de modalités uniques dans la colonne {columnn} excède
85                         le seuil de {max_modalities}. Aucun graphique en ligne généré.")
86     return
87
88     modality_data = {modality: [] for modality in all_modalities}
89     years = sorted(modality_counts.keys())
90
91     for modality in all_modalities:
92         for year in years:
93             count = modality_counts[year].get(modality, 0)
94             modality_data[modality].append(count)
95
96     df_modalities = pd.DataFrame(modality_data, index=years)
97
98     df_modalities.plot(kind='line', figsize=(15, 7), marker='o')
99     plt.title(f'Evolution of Distribution for column: {columnn}')
100    plt.xlabel('Years')
101    plt.ylabel('Proportion (%)')
102    plt.legend(title='Modalities', bbox_to_anchor=(1.05, 1), loc='upper left')
103    plt.tight_layout()
104    plt.show()
105
106 # Fonction de génération des graphiques par colonne et par dataset
107 def one_column_generate_plots(columnn, datasets):
108     create_boxplot_grid(columnn, datasets)
109     create_total_stacked_barplot(columnn, datasets)
110     create_lineplot_evolution(columnn, datasets)
111
112 # Fonction de génération des graphiques pour toutes les colonnes et tous les datasets
113 def all_columns_generate_plots(datasets):
114     all_columns = set()
115     for dataset in datasets:
116         for file_name, df in dataset.items():
117             logging.info(f'Chargement de {file_name}.')
118             all_columns.update(df.columns)
119
120     for columnn in all_columns:
121         one_column_generate_plots(columnn, datasets)
122
123 for datasets in dataframes:

```

```
123 all_columns_generate_plots(datasets)
```

#### Observations :

En raison de la longueur des résultats obtenus, ces derniers sont présentés plus en détails dans la partie annexe du présent rapport lorsqu'ils s'avèrent pertinents. Combinés aux éléments statistiques qui suivront, ils nous seront utiles notamment à l'occasion du nettoyage des données pour juger de l'intérêt de les conserver, regrouper, supprimer, ...

#### 1.3.2 Analyse des données brutes concaténées :

Pour une exploration plus poussée des données, il est procédé à une concaténation des fichiers CSV en un seul DataFrame pour chaque type de données. Il s'agit donc d'aller un peu plus loin dans la recherche d'éventuelles anomalies.

- **Concaténation des dataframes par rubrique :**

Notre première étape de prétraitement des données consiste à transformer les ensembles de données fragmentés en une ressource cohérente et plus facilement exploitable. Elle passe par plusieurs étapes :

- Le processus de transformation commence par l'initialisation d'un dictionnaire qui servira à stocker les données consolidées.
- Ensuite, le code parcourt systématiquement chaque préfixe (type de données) et ses fichiers associés. Pour chaque préfixe, il collecte tous les DataFrames correspondants, quelle que soit leur année d'origine.
- Une fois tous les DataFrames d'un type particulier rassemblés, le code les fusionne en un seul grand DataFrame. Cette opération est répétée pour chaque type de données, résultant en un ensemble de DataFrames consolidés, chacun représentant un aspect spécifique des données sur toute la période étudiée.

#### Code Python

```
1 # Initialisation d'un dictionnaire
2 concatenated_dfs = {}
3
4 # Parcourir chaque préfixe et liste de DataFrames
5 for prefix, df_list in zip(prefixes, dataframes):
6     # Liste pour stocker tous les DataFrames d'un préfixe
7     dfs_to_concat = []
8
9     # Parcourir chaque dictionnaire dans la liste de DataFrames
10    for dataset in df_list:
11        for _, df in dataset.items():
12            dfs_to_concat.append(df)
13
14    # Concaténer tous les DataFrames pour ce préfixe
15    if dfs_to_concat:
16        concatenated_df = pd.concat(dfs_to_concat, ignore_index=True)
17
18        # Ajouter le DataFrame concaténé au dictionnaire, nommé par le préfixe
19        concatenated_dfs[prefix] = concatenated_df
20
21 # Afficher des informations sur les DataFrames concaténés
22 for prefix, df in concatenated_dfs.items():
23     logging.info(f"Dataframe concaténé pour '{prefix}' : {len(df)} lignes, {df.shape[1]} colonnes")
```

#### Observations :

L'intérêt principal de cette approche est de simplifier considérablement la structure des données. Au lieu d'avoir de nombreux fichiers séparés par année, on obtient un ensemble réduit de DataFrames, chacun contenant toutes les données d'un type particulier sur l'ensemble de la période. Cette consolidation facilite grandement les analyses ultérieures, notamment des analyses statistiques plus complexes.

Au final, on obtient un aperçu de la taille de chaque DataFrame consolidé, qui offre une vision plus claire de la quantité de données disponibles pour chaque aspect étudié. Cette information est précieuse pour comprendre la portée des données à disposition.

Les dimensions des 4 DataFrames ainsi obtenus (soit 1 par rubrique) suivent :

- 'caracteristiques' : 1 176 873 lignes x 17 colonnes
- 'lieux' : 1 176 873 lignes x 19 colonnes
- 'usagers' : 2 636 377 lignes x 17 colonnes
- 'vehicules' : 2 009 395 lignes x 11 colonnes

- **Description des variables :**

Après avoir fourni leur brève description, chaque variable fait l'objet de statistiques, pour obtenir une vue d'ensemble de la structure et du contenu de chaque ensemble de données. Le tableau suivant reprend les éléments ayant servi de base à une meilleure compréhension de ces données, et les détails correspondants sont rendus disponibles en annexe du présent rapport :

TABLE 1: Description des variables et code associé


### 1.3.3 Analyse des données brutes fusionnées :

Pour continuer notre analyse, il s'avère nécessaire de fusionner les DataFrames et de réaliser d'autres tests statistiques.

- **Fonction pour fusionner les DataFrames :**

Code Python

```

1 def merge_dataframes(dataframes_list):
2     merged_dfs = {}
3     for prefix, df_list in zip(prefixes, dataframes_list):
4         if not df_list:
5             logging.warning(f"Aucun DataFrame trouvé pour le préfixe '{prefix}'")
6             continue
7
8         # Concaténer tous les DataFrames pour chaque préfixe
9         merged_df = pd.concat([list(df.values())[0] for df in df_list],
10                               ignore_index=True)
11         merged_dfs[prefix] = merged_df
12         logging.info(f"DataFrame fusionné pour '{prefix}': {len(merged_df)} lignes")
13
14     return merged_dfs
15
16 # Fusionner les DataFrames pour chaque préfixe
17 merged_dataframes = merge_dataframes(dataframes)
18
19 # Préparation pour la fusion finale
20 dataframes_to_merge = [
21     (merged_dataframes['usagers'], 'Num_Acc', 'left'),
22     (merged_dataframes['lieux'], 'Lat', 'right'),
23     (merged_dataframes['caracteristiques'], 'Vehicule', 'right'),
24     (merged_dataframes['vehicules'], 'Vehicule', 'left')
25 ]

```

```

21     (merged_dataframes['caracteristiques'], 'Num_Acc', 'left'),
22     (merged_dataframes['lieux'], 'Num_Acc', 'left'),
23     (merged_dataframes['vehicules'], ['Num_Acc', 'id_vehicule', 'num_veh'], 'left')
24 ]
25
26 # DataFrame de fusion
27 accidents = dataframes_to_merge[0][0]
28
29 # Boucle de jonction des DataFrames
30 for i in range(1, len(dataframes_to_merge)):
31     df_to_merge, merge_on, how = dataframes_to_merge[i]
32     accidents = pd.merge(accidents, df_to_merge, on=merge_on, how=how)
33
34 logging.info(f"DataFrame final 'accidents' créé avec {len(accidents)} lignes")

```

### Observations :

Les étapes du code créé sont les suivantes :

- D'abord, il concatène les données de même type provenant de différentes sources ou périodes. Pour chaque catégorie (usagers, caractéristiques, lieux, véhicules), il combine tous les DataFrames disponibles en un seul. Cette étape consolide les informations par type, regroupant par exemple toutes les données sur les usagers en un seul DataFrame.
- Ensuite, le code prépare la fusion finale de ces DataFrames consolidés. Il définit l'ordre dans lequel les différentes catégories de données seront combinées et spécifie les clés de jonction pour chaque fusion. Le processus de fusion commence alors avec le DataFrame des usagers comme base. Le code y ajoute successivement les informations des autres catégories : d'abord les caractéristiques des accidents, puis les données sur les lieux, et enfin les informations sur les véhicules impliqués. Chaque fusion est réalisée en utilisant des identifiants spécifiques, principalement le numéro d'accident ('Num\_Acc'), pour s'assurer que les informations sont correctement associées. Pour les véhicules, la fusion utilise plusieurs clés pour gérer la complexité supplémentaire de cette catégorie.
- Le type de fusion utilisé ('left') garantit que toutes les données des usagers soient conservées, même si certaines informations correspondantes manquent dans les autres catégories. Cela permet de préserver l'intégrité des données sur les personnes impliquées dans les accidents.
- Tout au long du processus, le code enregistre des informations sur le nombre de lignes dans chaque DataFrame fusionné et dans le DataFrame final. Ces logs permettent de suivre l'évolution de la taille des données et de vérifier que la fusion se déroule comme prévu.
- Le résultat final est un unique DataFrame nommé 'accidents', qui contient toutes les informations sur les accidents, les usagers impliqués, les lieux où ils se sont produits et les véhicules concernés. Ce DataFrame unifié facilite grandement les analyses à venir en fournissant une vue complète et intégrée de chaque accident dans un seul ensemble de données cohérent.

- **Tests statistiques :**

Afin d'appréhender l'orientation de nos recherches, quelques tests statistiques sont réalisés en vue d'identifier les facteurs pouvant avoir une relation significative avec la gravité des accidents de la route.

Voici la description linéaire de notre approche effectuée en plusieurs étapes :

- Nous avons créé un code qui prépare les données en nettoyant le DataFrame 'accidents'. Celui-ci remplace les valeurs problématiques par NaN et convertit certaines colonnes en format numérique. Il calcule également l'âge des personnes impliquées dans les accidents.
- Ensuite, le code définit une fonction d'analyse statistique nommée 'run\_statistical\_tests'. Cette fonction est conçue pour appliquer différents tests statistiques selon le type de variables. Pour les variables numériques, elle utilise des tests de corrélation de Spearman et de Pearson. Pour les variables catégorielles, elle applique le test du Chi-carré. De plus, elle effectue une analyse de variance (ANOVA) pour toutes les variables.
- Le code établit ensuite une liste exhaustive de caractéristiques à analyser. Ces caractéristiques couvrent divers aspects des accidents, incluant les informations sur les usagers, les véhicules, les conditions de l'accident, la route, le temps et la localisation.
- Après cette préparation, le code exécute les tests statistiques. Il applique la fonction 'run\_statistical\_tests' à toutes les caractéristiques sélectionnées, en utilisant 'grav' (la gravité de l'accident) comme variable cible.
- Enfin, le code organise et présente les résultats. Il compile tous les résultats des tests dans un DataFrame, les trie par valeur-p pour mettre en évidence les relations les plus significatives, et les affiche de manière complète et lisible.

## Code Python

```
1 # Prétraitement sommaire des données
2 df_merged = accidents.replace(['-1', 'N/A'], np.nan)
3 numeric_columns = ['grav', 'an_nais', 'vma', 'lartpc', 'larrou', 'jour', 'mois',
4     'hrmn', 'lat', 'long']
5 for col in numeric_columns:
6     df_merged[col] = pd.to_numeric(df_merged[col], errors='coerce')
7
8 df_merged['age'] = df_merged['an'] - df_merged['an_nais']
9
10 def run_statistical_tests(df, target_variable, features):
11     results = []
12
13     for feature in features:
14         # Supprimer les lignes avec des valeurs manquantes pour la caractéristique et la
15         # variable cible
16         df_clean = df[[feature, target_variable]].dropna()
17
18         # Vérifier s'il y a au moins deux valeurs uniques
19         if df_clean[feature].nunique() < 2 or df_clean[target_variable].nunique() < 2:
20             results.append({
21                 'Feature': feature,
22                 'Test': "Valeurs uniques",
23                 'P-value': np.nan,
24                 'Conclusion': "Insuffisant"
25             })
26             continue
27
28         if df_clean[feature].dtype in ['int64', 'float64']:
29
30             # Test de corrélation de Spearman
31             _, p_value = stats.spearmanr(df_clean[feature], df_clean[target_variable])
32             results.append({
33                 'Feature': feature,
34                 'Test': "Spearman",
35                 'P-value': p_value,
36                 'Conclusion': "Corr_significative" if p_value < 0.05 else
37                     "Corr_non_significative"
38             })
39
40             # Test de corrélation de Pearson
41             _, pearson_p = stats.pearsonr(df_clean[feature], df_clean[target_variable])
42             results.append({
43                 'Feature': feature,
44                 'Test': "Pearson",
45                 'P-value': pearson_p,
46                 'Conclusion': "Corr_significative" if pearson_p < 0.05 else
47                     "Corr_non_significative"
48             })
49
50         else:
51             # Test du Chi-carré
52             contingency_table = pd.crosstab(df_clean[feature], df_clean[target_variable])
53             if contingency_table.shape[0] > 1 and contingency_table.shape[1] > 1:
54                 _, chi2_p, dof, expected = stats.chi2_contingency(contingency_table)
55                 results.append({
56                     'Feature': feature,
57                     'Test': "Chi2",
58                     'P-value': chi2_p,
59                     'Conclusion': "Ass_significative" if chi2_p < 0.05 else
60                         "Ass_non_significative"
61                 })
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
737
738
739
739
740
741
742
743
744
745
745
746
747
747
748
749
749
750
751
752
753
754
755
755
756
757
757
758
759
759
760
761
762
763
764
764
765
766
766
767
767
768
768
769
769
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1
```

```

56         })
57
58     # Test ANOVA à un facteur (pour toutes les variables)
59     groups = [group for _, group in df_clean.groupby(feature)[target_variable]]
60     if len(groups) > 1:
61         try:
62             _, f_p = stats.f_oneway(*groups)
63             results.append({
64                 'Feature': feature,
65                 'Test': "ANOVA",
66                 'P-value': f_p,
67                 'Conclusion': "Diff_significative" if f_p < 0.05 else
68                             "Diff_non_significative"
69             })
70         except ValueError:
71             # Si ANOVA échoue, on ignore simplement ce test
72             pass
73
74     return pd.DataFrame(results)
75
76 # Liste des caractéristiques à tester
77 features_to_test = [
78     'age', 'sexe', 'catu', 'place', 'locp', 'etatp', 'actp', # Caractéristiques des
79     usagers
80     'motor', 'vma', # Caractéristiques des véhicules
81     'lum', 'atm', 'col', 'secu', 'secu1', 'secu2', 'secu3', 'choc', 'trajet', 'catv',
82     'obs', 'obsm', 'manv', 'occutc', # Conditions de l'accident
83     'catr', 'circ', 'vosp', 'prof', 'plan', 'surf', 'infra', 'situ', 'nbv', 'pr', 'pr1',
84     'lartpc', 'larrouut', # Caractéristiques de la route
85     'an', 'an_nais', 'hrmn', 'jour', 'mois', # Caractéristiques temporelles
86     'adr', 'voie', 'v1', 'v2', 'env1', 'agg', 'int', 'com', 'dep', 'gps', 'lat', 'long'
87     # Localisation
88 ]
89
90 # Exécution des tests
91 test_results = run_statistical_tests(df_merged, 'grav', features_to_test)
92
93 pd.set_option('display.max_rows', None) # Affiche toutes les lignes
94 pd.set_option('display.max_columns', None) # Affiche toutes les colonnes
95 pd.set_option('display.width', None) # Utilise toute la largeur de l'écran
96 pd.set_option('display.max_colwidth', None) # Affiche le contenu complet des cellules
97
98 # Affichage des résultats
99 print("Résultats des tests statistiques:")
100 print(test_results.sort_values('P-value'))
101 pd.reset_option('display.max_rows')
102 pd.reset_option('display.max_columns')
103 pd.reset_option('display.width')
104 pd.reset_option('display.max_colwidth')

```

#### Types de Tests Utilisés :

- Spearman (Corrélation de Spearman) : Mesure la corrélation monotone entre deux variables. Utilisé pour des données ordinaires ou lorsque les relations ne sont pas linéaires.
- Pearson (Corrélation de Pearson) : Mesure la corrélation linéaire entre deux variables continues.
- ANOVA (Analyse de la Variance) : Teste les différences entre les moyennes de plusieurs groupes pour une variable continue.
- Chi2 (Test du Chi-carré d'indépendance) : Utilisé pour déterminer s'il existe une association significative entre deux variables catégorielles.

#### P-Values et Significativité :

Une p-value proche de 0 (par exemple, 0.000000e+00) indique que le test statistique a trouvé une relation significative (corrélation, différence, ou association) entre les variables testées.

Une p-value supérieure au seuil de 0,05 signifie qu'il n'y a pas de relation significative.

**Interprétation des Résultats des Tests :**

- Corr\_significative : Indique une corrélation significative entre la variable d'intérêt et une autre variable.
- Diff\_significative : Indique une différence significative entre les groupes de la variable testée.
- Ass\_significative : Indique une association significative entre deux variables catégorielles.
- Corr\_non\_significative / Diff\_non\_significative : Indique l'absence de corrélation ou de différence significative.

Voici les résultats des tests statistiques :

---

---

Suite de la page précédente

---

---

Suite sur la page suivante

---

---

## Résultats Significatifs

### — CORRÉLATIONS ET DIFFÉRENCES SIGNIFICATIVES :

Les variables telles que **age**, **obs**, **catv**, **trajet**, **choc**, **secu1**, **secu2**, **sexe**, **manv**, et **etatp** ont des résultats de tests Spearman, Pearson, ou ANOVA avec des p-values très proches de zéro (0.000000e+00). Cela signifie qu'il existe des corrélations significatives ou des différences significatives entre ces variables et la variable cible ou entre elles. Par exemple, pour la variable age, les tests de corrélation de Spearman et de Pearson ainsi que le test ANOVA ont tous une p-value de 0.000000e+00, ce qui indique une forte corrélation et des différences significatives liées à l'âge.

### — ASSOCIATIONS SIGNIFICATIVES (Chi2) :

Les variables **voie**, **pr**, **com**, **dep**, **gps**, **v2**, **actp**, et **adr** montrent des associations significatives avec des p-values de 0.000000e+00 ou proches de zéro dans les tests du Chi2. Cela suggère que ces variables catégorielles sont statistiquement associées entre elles ou avec d'autres variables d'intérêt.

### — RÉSULTATS NON SIGNIFICATIFS :

Certaines variables, comme **vosp**, **jour**, **agg**, **voie**, **adr**, **lat**, et **long**, ont des résultats de tests avec des p-values élevées (par exemple, 2.143277e-01, 3.827203e-01, jusqu'à 1.000000e+00). Cela signifie que ces tests n'ont pas trouvé de corrélation, de différence, ou d'association statistiquement significative.

## 1.4 Constats préalables avant le preprocessing des données :

L'exploration approfondie des données nous a permis d'avoir une meilleure compréhension de notre jeu de données. Nous avons examiné la structure des données, identifié les principales variables, visualisé les distributions, et effectué des tests statistiques préliminaires.

Ces analyses exploratoires ont révélé plusieurs points importants :

- La complexité des informations disponibles dans les différentes rubriques (caractéristiques, lieux, usagers, véhicules).
- La présence de valeurs manquantes, aberrantes ou codées de manière inconsistante dans certaines variables.
- Des relations statistiquement significatives entre plusieurs variables et la gravité des accidents.
- La nécessité de traiter les problèmes de qualité des données et de préparer celles-ci pour une modélisation efficace.

Ces constats mettent en évidence la nécessité d'un prétraitement des données avant toute modélisation fiable. Le prochain chapitre sera donc consacré au nettoyage des données en vue de notre première itération. Cette étape nous permettra d'évaluer l'efficacité du prétraitement et d'identifier les améliorations potentielles pour optimiser notre approche de modélisation.

## 2 Premier prétraitement des données et son application modélisée

### 2.1 Etape 1 – Restructuration des dataframes :

#### 2.1.1 Création de fonctions spécifiques

La fonction 'convert\_dtypes' permet de corriger les variations de type constatées dans certaines colonnes des DataFrames 'df'. Elle parcourt les colonnes du DataFrame, tente de les convertir aux types spécifiés dans le dictionnaire de référence 'reference\_dtypes', et gère les erreurs de conversion. La fonction retourne le DataFrame avec les types de données ajustés, tout en enregistrant les éventuels échecs de conversion.

Code Python

```
1 # Conversion des types de données
2 def convert_dtypes(df, reference_dtypes):
3     for col in df.columns:
4         if col in reference_dtypes:
5             try:
6                 df.loc[:, col] = df[col].astype(reference_dtypes[col])
7             except ValueError as e:
8                 logging.error(f"Erreur lors de la conversion de la colonne {col} en type
9                               {reference_dtypes[col]}: {e}")
9     return df
```

La fonction 'extract\_reference\_structure' crée un dictionnaire 'reference\_structures' à partir d'une liste de 'dataframes'. Pour chaque paire prefix-dataframe, elle extrait les 'dtypes' du dernier DataFrame de la liste, les associe au 'prefix' correspondant dans 'reference\_structures'. Le résultat est un dictionnaire où chaque clé 'prefix' est associée aux types de données de son dernier DataFrame.

Code Python

```
1 # Extraction de la structure de référence
2 def extract_reference_structure(dataframes):
3     reference_structures = {}
4     for prefix, df_list in zip(prefixes, dataframes):
5         if df_list:
6             last_df = list(df_list[-1].values())[0]
7             reference_structures[prefix] = last_df.dtypes.to_dict()
8     return reference_structures
```

La fonction 'preprocess' prend en entrée un DataFrame 'df' et son 'prefix'. Elle applique une série de transformations au DataFrame selon la valeur de son 'prefix' et le retourne nettoyé et standardisé.

#### Code Python

```

1 # Prétraitement des données
2 def preprocess(df, prefix):
3     if prefix == 'caracteristiques':
4         if 'Accident_Id' in df.columns:
5             df = df.rename(columns={'Accident_Id': 'Num_Acc'})
6         if 'an' in df.columns:
7             df.loc[:, 'an'] = df['an'].apply(lambda x: x + 2000 if x < 2000 else x)
8         if 'hrmn' in df.columns:
9             df.loc[:, 'hrmn'] = df['hrmn'].apply(lambda x:
10                 f'{str(x).zfill(4)[2:]}' if str(x).zfill(4)[2:] != '' else '')
11        for col in ['lum', 'int', 'atm', 'col']:
12            if col in df.columns:
13                df[col] = df[col].fillna(-1)
14        df = df.drop(columns=['adr', 'lat', 'long'], errors='ignore')
15    elif prefix == 'lieux':
16        for col in ['circ', 'vosp', 'prof', 'pr', 'pri', 'plan', 'surf', 'infra',
17                    'situ']:
18            if col in df.columns:
19                df[col] = df[col].fillna(-1)
20        if 'lartpc' in df.columns:
21            df['lartpc'] = df['lartpc'].replace(0, -1).fillna(-1)
22        if 'vma' in df.columns:
23            df['vma'] = df['vma'].apply(lambda x: -1 if pd.isna(x) or x > 130 else x)
24        df = df.drop(columns=['voie', 'v1', 'v2', 'larrout'], errors='ignore')
25    elif prefix == 'usagers':
26        for col in ['place', 'catu', 'grav', 'sexe', 'trajet', 'secu1', 'secu2',
27                    'secu3', 'locp', 'actp', 'etatp']:
28            if col in df.columns:
29                df[col] = df[col].fillna(-1)
30        if 'catu' in df.columns:
31            df['catu'] = df['catu'].replace(4, -1)
32        if 'an_nais' in df.columns:
33            df['an_nais'] = df['an_nais'].apply(lambda x: pd.NA if x < 1900 else x)
34    elif prefix == 'vehicules':
35        for col in ['senc', 'obs', 'obsm', 'choc', 'manv', 'motor']:
36            if col in df.columns:
37                df[col] = df[col].fillna(-1)
38        if 'catv' in df.columns:
39            df['catv'] = df['catv'].fillna(0)
40        df = df.drop(columns=['occutc'], errors='ignore')
41        for col in df.columns:
42            if df[col].dtype == np.float64 or df[col].dtype == np.int64:
43                df.loc[:, col] = df[col].fillna(-1)
44        if 'lartpc' in df.columns:
45            df.loc[:, 'lartpc'] = df['lartpc'].replace(0, -1).fillna(-1)
46        if 'catu' in df.columns:
47            df.loc[:, 'catu'] = df['catu'].replace(4, -1)
48        if 'an_nais' in df.columns:
49            df.loc[:, 'an_nais'] = df['an_nais'].apply(lambda x: pd.NA if x < 1900 else
50                                              x)
51    return df

```

La fonction 'preprocess\_datasets' permet d'appliquer la fonction 'preprocess' à chaque DataFrame 'df' contenu dans chaque 'df\_dict' de chaque 'df\_list', en utilisant le 'prefix' associé. Les DataFrames traités sont ensuite remplacés dans 'df\_dict' sous leur 'file\_name' d'origine. La fonction retourne les 'dataframes' prétraités, en conservant leur structure initiale.

### Code Python

```
1 # Prétraitement des ensembles de données
2 def preprocess_datasets(dataframes):
3     for prefix, df_list in zip(prefixes, dataframes):
4         for df_dict in df_list:
5             for file_name, df in df_dict.items():
6                 df = preprocess(df, prefix)
7                 df_dict[file_name] = df
8
9     return dataframes
```

La fonction 'harmonize\_dataframes' vise à harmoniser les colonnes et le type de données pour chaque data-frame, selon une structure de référence 'reference\_structures' définie par la fonction 'convert\_dtypes'.

### Code Python

```
1 # Harmonisation des dataframes
2 def harmonize_dataframes(dataframes, reference_structures):
3     harmonized_dataframes = []
4     for prefix, df_list in zip(prefixes, dataframes):
5         reference_dtypes = reference_structures.get(prefix, {})
6         harmonized_group = []
7         for df_dict in df_list:
8             for file_name, df in df_dict.items():
9                 df = df[[col for col in df.columns if col in reference_dtypes]]
10                df = convert_dtypes(df, reference_dtypes)
11                harmonized_group.append({file_name: df})
12            harmonized_dataframes.append(harmonized_group)
13
14    return harmonized_dataframes
```

La fonction 'compare\_structures' compare les structures des dataframes harmonisés (résultant de la fonction 'harmonize\_dataframes') par rapport à la structure de référence (émanant de la fonction 'convert\_dtypes'), afin de repérer les différences pouvant exister en termes de colonnes et de types de données.

### Code Python

```
1 # Comparaison des structures
2 def compare_structures(harmonized_dataframes, reference_structures):
3     for prefix, df_list in zip(prefixes, harmonized_dataframes):
4         reference_dtypes = reference_structures.get(prefix, {})
5         for df_dict in df_list:
6             for file_name, df in df_dict.items():
7                 df_dtypes = df.dtypes.to_dict()
8                 missing_cols = set(reference_dtypes.keys()) - set(df_dtypes.keys())
9                 extra_cols = set(df_dtypes.keys()) - set(reference_dtypes.keys())
10                diff_types = {col: (reference_dtypes[col], df_dtypes[col]) for col in
11                               df_dtypes if col in reference_dtypes and reference_dtypes[col] != df_dtypes[col]}
12                table_data = [
13                    ["Colonnes manquantes", ", ".join(missing_cols)],
14                    ["Colonnes supplémentaires", ", ".join(extra_cols)],
15                    ["Différences de types", ", ".join([f"{{col}} (réf:
16                                     {reference_dtypes[col]}, fichier: {df_dtypes[col]})" for col in
17                                     diff_types])]
18                ]
19                table = tabulate(table_data, headers=["Type de différence", "Colonnes"],
20                                tablefmt="grid")
```

17

```
logging.info(f"Comparaison pour le fichier {file_name}:\n{table}")
```

La fonction 'concat\_harmonized\_dataframes' crée une liste de dataframes concaténés 'concatenated\_dataframes', en vue d'une utilisation ultérieure.

Code Python

```
1 # Liste de dataframes concaténés
2 def concat_harmonized_dataframes(dataframes):
3     concatenated_dataframes = []
4
5     for df_list in dataframes:
6
7         df_only_list = [list(d.values())[0] for d in df_list]
8         concatenated_df = pd.concat(df_only_list, ignore_index=True)
9         concatenated_dataframes.append(concatenated_df)
10
11 return concatenated_dataframes
```

La fonction 'merge\_dataframes' fusionne l'ensemble des dataframes concaténés ('concatenated\_dataframes') en considération des spécificités mentionnées au niveau de chaque jointure :

Code Python

```
1 # Fusion des dataframes
2 def merge_dataframes(concatenated_dataframes):
3     df_caracteristiques, df_lieux, df_usagers, df_vehicules = concatenated_dataframes
4
5     merged_df = pd.merge(df_caracteristiques, df_lieux, on='Num_Acc', how='inner')
6     merged_df = pd.merge(merged_df, df_usagers, on='Num_Acc', how='inner')
7     merged_df = pd.merge(merged_df, df_vehicules, on=['Num_Acc', 'id_vehicule',
8             'num_veh'], how='inner')
9
10 return merged_df
```

Une fois la définition des fonctions achevées, celles-ci sont exécutées pour mettre en œuvre les étapes d'harmonisation des dataframes.

### 2.1.2 Considérations préalables

La modification de la codification de la variable cible 'grav' en 2019 représente une cause majeure de l'hétérogénéité structurelle des dataframes annuels, marquant une rupture significative dans la continuité des données. Pour assurer une cohérence optimale de notre étude, nous avons décidé de ne garder que les données de la période 2019-2022. Ce choix est conforté par l'introduction en 2019 de nouvelles variables, telles que 'vma' (vitesse maximale autorisée), et par une restructuration générale des données (exemple : 'secu1', 'secu2', etc.). Cette approche permet de travailler avec des données plus récentes et plus homogènes, facilitant l'uniformisation nécessaire pour les prochaines étapes de prétraitement et de modélisation.

### 2.1.3 Lancement du processus

Le code suivant prépare les données des accidents routiers pour analyse. Il commence par définir la plage des années considérées (2019-2022) et les catégories de données (caractéristiques, lieux, usagers, véhicules). Ensuite, il charge ces données au moyen des fonctions préalablement définies.

#### Code Python

```
1 # Définition des paramètres
2 years = list(range(2019, 2023))
3 prefixes = ['caracteristiques', 'lieux', 'usagers', 'vehicules']
4
5 # Chargement et prétraitement des données
6 dataframes = load_datasets(prefixes, years)
7 dataframes = preprocess_datasets(dataframes)
8
9 # Harmonisation des dataframes en fonction d'une structure de référence
10 reference_structures = extract_reference_structure(dataframes)
11 harmonized_dataframes = harmonize_dataframes(dataframes, reference_structures)
```

#### 2.1.4 Contrôles avant fusion

Le code suivant fournit un aperçu rapide du nombre de datasets chargés pour chaque catégorie de données (caractéristiques, lieux, usagers, véhicules) et indique le nombre total de datasets chargés pour toutes les catégories confondues :

#### Code Python

```
1 # Vérification des datasets chargés
2 for prefix, df_list in zip(prefixes, dataframes):
3     logging.info(f'{prefix}: {len(df_list)} datasets chargés.')
4
5 logging.info(f'Total datasets chargés: {sum(len(dfs) for dfs in dataframes)}.'
```

Ce code parcourt un dictionnaire pour examiner les structures de référence définies pour chaque catégorie de données, et log chacune d'elles :

#### Code Python

```
1 # Vérification des structures de référence
2 for prefix, structure in reference_structures.items():
3     logging.info(f'Structure de référence pour {prefix}: {structure}'')
```

La fonction 'compare\_structures' examine chaque dataframe harmonisé et le compare à la structure de référence correspondante. Elle permet de détecter les anomalies potentielles résultant du processus d'harmonisation, telles que des colonnes manquantes, des colonnes inattendues, ou des incohérences dans les types de données.

#### Code Python

```
1 # Comparaison des dataframes harmonisés aux structures de référence
2 compare_structures(harmonized_dataframes, reference_structures)
```

Voici un extrait des résultats observés :

2024-09-10 12 :04 :02,235 - INFO - Comparaison pour le fichier data/raw/usagers-2019.csv :

Type de différence	Colonnes
Colonnes manquantes	id_usager
Colonnes supplémentaires	
Défauts de types	an_nais (réf : float64, fichier : int64)

2024-09-10 12 :04 :02,246 - INFO - Comparaison pour le fichier data/raw/usagers-2020.csv :

Type de différence	Colonnes
Colonnes manquantes	id_usager
Colonnes supplémentaires	
Défauts de types	an_nais (réf : float64, fichier : int64)

Grâce à l'affichage des premières lignes de chaque dataframe harmonisé, le code suivant permet de vérifier visuellement si toutes les colonnes attendues sont présentes dans chaque dataframe et de repérer si les types de données semblent corrects et cohérents.

Code Python

```
1 # Affichage des premières lignes des dataframes harmonisés
2 for group in harmonized_dataframes:
3     for df_dict in group:
4         for file_name, df in df_dict.items():
5             print(f"Head of harmonized dataframe {file_name}:")
6             display(df.head())
```

Le code suivant met en mémoire la fonction 'concat\_harmonized\_dataframes' destinée à fusionner tous les dataframes harmonisés de chaque catégorie (préfixe) en un seul dataframe par catégorie.

Code Python

```
1 # Listing des dataframes harmonisés
2 concatenated_dataframes = concat_harmonized_dataframes(harmonized_dataframes)
```

Le code suivant appelle le précédent en créant une boucle qui parcourt les dataframes concaténés pour associer chaque dataframe à son préfixe. Le code affiche la dimension de chaque ensemble de données consolidé (nombre de lignes et de colonnes) pour chaque dataframe concaténé.

Code Python

```
1 # Affichage des dimensions des dataframes concaténés
2 for prefix, df in zip(prefixes, concatenated_dataframes):
3     logging.info(f"{prefix}: {df.shape}")
```

#### Observations :

L'exécution progressive de cet ensemble de boucles et de fonctions a permis de vérifier le bon déroulement du processus de chargement et de structuration des données avant d'aboutir à la fusion des dataframes.

#### 2.1.5 Fusion des dataframes harmonisés

Code Python

```
1 # Vérification de la dimension des dataframes concaténés après leur fusion
2 final_merged_df = merge_dataframes(concatenated_dataframes)
3 logging.info(f"DataFrame final fusionné: {final_merged_df.shape}")
```

#### Observations :

Les résultats obtenus appellent de nouvelles vérifications.

## 2.2 Etape 2 - Prétraitement après fusion des dataframes :

### 2.2.1 Contrôles après fusion

La fonction 'check\_nan\_presence' effectue un contrôle sur le dataframe final fusionné et calcule le pourcentage des valeurs manquantes (NaN) pour chaque colonne concernée. Les résultats sont affichés pour fournir un aperçu de la qualité des données après fusion.

Code Python

```
1 # Contrôle des valeurs nulles après fusion
2 def check_nan_presence(df):
3     nan_columns = df.columns[df.isna().any()].tolist()
4     if nan_columns:
5         nan_proportions = df[nan_columns].isna().mean() * 100
6         for col, prop in nan_proportions.items():
7             logging.info(f"Colonne '{col}' contient {prop:.2f}% de NaN.")
8     else:
9         logging.info("Aucune colonne ne contient de NaN.")
10
11 check_nan_presence(final_merged_df)
```

La fonction 'check\_nunique' fournit le nombre de valeurs uniques dans chaque colonne et affiche ces informations pour aider à comprendre la structure et la complexité du dataframe final fusionné.

Code Python

```
1 # Contrôle des valeurs uniques après fusion
2 def check_nunique(df):
3     unique_proportions = df.nunique()
4     for col, prop in unique_proportions.items():
5         logging.info(f"Colonne '{col}' a {prop} valeurs uniques.")
6
7 check_nunique(final_merged_df)
```

#### Observations :

Ces contrôles successifs ont permis de relever les valeurs sur lesquelles il sera utile de revenir :

- au niveau des valeurs nulles pour les colonnes 'an\_naiss' et 'id\_usager' et,
- au niveau des valeurs uniques qui restent trop élevées pour certaines variables.

Les éléments concernés ont été traduits dans la fonction preprocessing\_final\_dataframe qui suit.

### 2.2.2 Crédation de fonctions spécifiques

La fonction suivante permet de réaliser un prétraitement complémentaire des données sur certains aspects spécifiques :

Code Python

```
1 # Prétraitement final du dataframe
2 def preprocessing_final_dataframe(df):
3     df = df.drop(columns=['id_usager', 'Num_Acc', 'com', 'id_vehicule',
4                           'num_veh', 'lartpc'])
5
6     if 'hrmn' in df.columns:
7         df['hour'] = df['hrmn'].str[:2].astype(int)
8         df = df.drop(columns=['hrmn'])
```

```

8
9     if 'an_nais' in df.columns:
10         mode_an_nais = df['an_nais'].mode()[0]
11         df['an_nais'] = df['an_nais'].fillna(mode_an_nais).astype(int)
12
13     return df

```

La fonction suivante transforme les données catégorielles du dataframe final en variables numériques au moyen d'un codage one-hot :

Code Python

```

1 # Encodage du dataframe
2 def encode_dataframe(df):
3     dummy_columns = ['lum', 'agg', 'int', 'atm', 'col', 'catr', 'circ', 'prof', 'place',
4                       'catu', 'sexe', 'trajet', 'secu1', 'secu2', 'secu3', 'locp', 'actp', 'etatp',
5                       'senc', 'catv', 'obs', 'obsm', 'choc', 'manv', 'motor', 'plan',
6                       'surf', 'an', 'infra', 'dep', 'situ', 'vosp']
7     df = pd.get_dummies(df, columns=dummy_columns, drop_first=True)
8
9     return df

```

### 2.2.3 Lancement du processus

Le code suivant applique la fonction 'preprocessing\_final\_dataframe' (définie en amont) au dataframe pour réaliser un nettoyage résiduel de points spécifiques, avant de vérifier notamment le résultat obtenu sur l'amplitude horaire des données.

Code Python

```

1 # Lancement du processus de preprocessing final
2 final_merged_df = preprocessing_final_dataframe(final_merged_df)
3
4 # Vérification de l'amplitude horaire
5 min_hour = final_merged_df['hour'].min()
6 max_hour = final_merged_df['hour'].max()
7 print(f"Min hour: {min_hour}, Max hour: {max_hour}")

```

Le code suivant finalise la préparation des données, assurant une conversion des colonnes de type 'object' en type 'int64', pour que toutes les variables du dataframe aient un format approprié pour la modélisation. La sauvegarde du fichier CSV permet de conserver une version propre et structurée des données pour une utilisation ultérieure.

Code Python

```

1 # Lancement du processus d'encodage des variables catégorielles
2 final_merged_df = encode_dataframe(final_merged_df)
3
4 # Repérage des colonnes 'objet' restantes
5 object_columns = final_merged_df.select_dtypes(include=['object']).columns
6 print("Colonnes objet restantes:", object_columns)
7 print("Shape du dataframe:", final_merged_df.shape)
8
9 # Boucle de modification des colonnes 'objet' restantes
10 for column in object_columns:
11     try:
12         final_merged_df[column] = final_merged_df[column].astype('int64')
13     except ValueError:
14         final_merged_df = final_merged_df[pd.to_numeric(final_merged_df[column],

```

```

        errors='coerce').notnull()]
15 final_merged_df[column] = final_merged_df[column].astype('int64')
16
17 # Affichage de la nouvelle dimension du dataframe
18 print("Nouveau shape du dataframe:", final_merged_df.shape)
19 display(final_merged_df.columns)
20
21 # Chemin de sauvegarde CSV
22 PATH_TO_CSVS_PROCESSED = 'data/processed/data_init.csv'
23 final_merged_df.to_csv(PATH_TO_CSVS_PROCESSED)

```

## 2.3 Etape 3 - Modélisation 1

### 2.3.1 Implémentation d'un premier modèle

Sur la base du fichier obtenu, il est procédé à une 1ère modélisation de type classique, à savoir une régression logistique. Le choix de ce modèle se justifie entre autres par les raisons suivantes :

- il est adapté à la nature binaire de la variable cible (accident grave ou non grave),
- il est capable de gérer les nombreuses variables catégorielles présentes dans le jeu de données,
- il offre une bonne base pour servir de référence à des modèles plus complexes, etc.

Le code suivant implémente donc un pipeline complet d'apprentissage automatique pour notre problème de classification. Il charge des données, les prétraite, puis effectue une recherche sur grille des hyperparamètres avec validation croisée, en utilisant une classe personnalisée qui permet de suivre la progression et de sauvegarder les résultats intermédiaires. Le processus inclut la standardisation des variables, la séparation des données en ensembles d'entraînement et de test, et l'optimisation des hyperparamètres tels que la force de régularisation, la pénalité et le ratio 11.

Code Python

```

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from sklearn.preprocessing import StandardScaler
5 from sklearn.model_selection import train_test_split, KFold, cross_validate,
   GridSearchCV, ParameterGrid
6 from sklearn.linear_model import LogisticRegression
7 from sklearn.metrics import accuracy_score, classification_report
8 import joblib
9 import time
10
11 # Chargement et prétraitement des données
12 final_merged_df = pd.read_csv('data/processed/data_init.csv', index_col=0)
13
14 X = final_merged_df.drop(columns=['grav'])
15 y = final_merged_df['grav']
16
17 # Standardisation des variables numériques
18 dummy_columns = ['lum', 'agg', 'int', 'atm', 'col', 'catr', 'circ', 'prof', 'place',
   'catu', 'sexe', 'trajet', 'secul1', 'secul2', 'secul3', 'locp', 'actp', 'etatp',
   'senc', 'catv', 'obs', 'obsm', 'choc', 'manv', 'motor', 'plan',
   'surf', 'an', 'infra', 'dep', 'situ', 'vosp']
19
20 columns_to_scale = [col for col in X.columns if col not in dummy_columns]
21
22 scaler = StandardScaler()
23 X_scaled = X.copy()
24 X_scaled[columns_to_scale] = scaler.fit_transform(X[columns_to_scale])
25
26 # Création d'un ensemble d'entraînement et d'un ensemble test
27 X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,

```

```

        random_state=12)
28
29 # Définition du classificateur
30 LR = LogisticRegression(random_state=123, tol=1e-3, solver='saga')
31
32 # Validation croisée
33 cv3 = KFold(n_splits=3, random_state=111, shuffle=True)
34
35 # Définition de l'espace de recherche des hyperparamètres
36 param_grid = [
37     {'C': [0.1, 1, 10], 'penalty': ['l1'], 'max_iter': [1000]},
38     {'C': [0.1, 1, 10], 'penalty': ['l2'], 'max_iter': [1000]},
39     {'C': [0.1, 1, 10], 'penalty': ['elasticnet'], 'max_iter': [1000], 'l1_ratio':
40         np.linspace(0, 1, 6)}
41 ]
42
43 # Création d'un GridSearchCV personnalisé pour sauvegarder les résultats partiels
44 class GridSearchWithProgress(GridSearchCV):
45     def fit(self, X, y=None, **fit_params):
46         n_candidates = sum(len(ParameterGrid(grid)) for grid in self.param_grid)
47         n_completed = 0
48
49         results = []
50
51         for params in ParameterGrid(self.param_grid):
52             start_time = time.time()
53             self.estimator.set_params(**params)
54             cv_results = cross_validate(self.estimator, X, y, cv=self.cv,
55                 scoring=self.scoring, return_train_score=True)
56             end_time = time.time()
57
58             result = {
59                 'params': params,
60                 'mean_test_score': cv_results['test_score'].mean()
61             }
62
63             results.append(result)
64
65             n_completed += 1
66
67             # Enregistrer les résultats partiels
68             joblib.dump(results, 'gridcv_resultsLR.joblib')
69
70             # Affichage des résultats partiels
71             print(f"Résultat partiel {n_completed}/{n_candidates}:")
72             print(f"Paramètres: {params}")
73             print(f"Score moyen: {cv_results['test_score'].mean():.4f} (+/-"
74                 f" {cv_results['test_score'].std():.4f})")
75             print(f"Temps de fit: {end_time - start_time:.2f} secondes")
76             print()
77
78             self.cv_results_ = results
79             best_index = max(range(len(results)), key=lambda i:
80                 results[i]['mean_test_score'])
81             self.best_params_ = results[best_index]['params']
82             self.best_score_ = results[best_index]['mean_test_score']
83
84             # Définition du meilleur estimateur
85             self.best_estimator_ = self.estimator.set_params(**self.best_params_)
86             self.best_estimator_.fit(X, y)

```

```

84         return self
85
86
87 # Exécution du GridSearchCV avec contrôle de l'état d'avancement
88 grid_search = GridSearchWithProgress(LR, param_grid, cv=cv3, n_jobs=-1, verbose=2)
89 grid_search.fit(X_train, y_train)
90
91 # Chargement des résultats enregistrés
92 results = joblib.load('gridcv_resultsLR.joblib')

```

Voici les 3 premiers résultats obtenus :

Résultat partiel 1/24 :  
Paramètres : 'C' : 0.1, 'max\_iter' : 1000, 'penalty' : 'l1'  
Score moyen : 0.6736 (+/- 0.0028)  
Temps de fit : 498.71 secondes

Résultat partiel 2/24 :  
Paramètres : 'C' : 1, 'max\_iter' : 1000, 'penalty' : 'l1'  
Score moyen : 0.6737 (+/- 0.0025)  
Temps de fit : 573.75 secondes

Résultat partiel 3/24 :  
Paramètres : 'C' : 10, 'max\_iter' : 1000, 'penalty' : 'l1'  
Score moyen : 0.6736 (+/- 0.0025)  
Temps de fit : 632.74 secondes

### 2.3.2 Évaluation des performances du modèle

A la suite des résultats obtenus, il est procédé à l'évaluation des performances du modèle.

Code Python

```

1 # Affichage des meilleurs paramètres et du meilleur score
2 print("Meilleurs paramètres:", grid_search.best_params_)
3 print("Meilleur score:", grid_search.best_score_)

4
5 # Utilisation du meilleur modèle
6 best_model = grid_search.best_estimator_
7 y_pred_best = best_model.predict(X_test)
8 print("Accuracy du meilleur modèle:", accuracy_score(y_test, y_pred_best))

```

Voici l'affichage correspondant :

Meilleurs paramètres : 'C' : 0.1, 'l1\_ratio' : 0.2, 'max\_iter' : 1000, 'penalty' : 'elasticnet'  
Meilleur score : 0.6738685114087616  
Accuracy du meilleur modèle : 0.6719810449379543

Le rapport de classification permet de noter les différentes métriques du meilleur modèle de prédiction retenu.

Code Python

```

1 # Rapport de classification détaillé pour le meilleur modèle
2 print("\nRapport de classification du meilleur modèle:\n", classification_report(y_test,
   y_pred_best, zero_division=0))

```

Voici l'affichage correspondant :

Rapport de classification du meilleur modèle :

	precision	recall	f1-score	support
1	0.70	0.82	0.76	10221
2	0.63	0.06	0.12	572
3	0.53	0.39	0.45	3631
4	0.67	0.66	0.66	10477
accuracy			0.67	24901
macro avg	0.63	0.48	0.50	24901
weighted avg	0.66	0.67	0.66	24901

### 2.3.3 Analyse des résultats

Du point de vue de la performance globale, l'accuracy de 67% indique que le modèle est plutôt satisfaisant, mais il y a encore une marge d'amélioration significative. Le modèle performe de manière cohérente entre la validation croisée (67.39%) et l'ensemble de test (67.20%), ce qui suggère une bonne généralisation.

Par contre, la performance par classe du modèle est clairement insatisfaisante : d'abord parce que les classes sont mal réparties entre elles (sur-représentation des classes 1 et 4), ensuite parce que la classe 2 relative aux accidents graves est très mal prédite (recall : 0.06 et F1-score : 0.12).

Au-delà du rééquilibrage nécessaire des classes, il est important de noter que la régression logistique est un modèle linéaire. De ce fait, elle peut ne pas capturer des relations complexes ou non linéaires de nos données. Il pourrait donc être intéressant d'essayer des algorithmes non linéaires comme les forêts aléatoires ou les gradient boosting machines pour aller plus loin. Aussi, combiner plusieurs modèles peut permettre d'améliorer la robustesse des prédictions.

## 3 Deuxième prétraitement des données et ses applications modélisées

Le premier prétraitement des données nous a permis d'effectuer une restructuration initiale des dataframes, de créer des variables de base et d'éliminer certaines variables non pertinentes. Cependant, la modélisation qui en a résulté nous a révélé des insuffisances au niveau des métriques obtenues. C'est pourquoi nous en concluons que certaines variables nécessitent une catégorisation plus fine, pour que les relations complexes entre les variables soient mieux exploitées et que le déséquilibre des classes puisse être moins persistant.

Dans cet objectif, nous appréhendons les choses sous un tout nouvel angle qui prend nécessairement en considération les résultats observés jusqu'ici pour parvenir à leur amélioration. Également, nous nous appuierons sur les observations suivantes pour définir la structuration ultérieure de nos dataframes.

### 3.1 Base de structuration des dataframes :

#### 3.1.1 Structure pour 'caracteristiques' :

Colonne	Problématique	Modification/Suppression
'Num_Acc' : dtype('int64')	Nom de variable qui change en 2022 + Trop de valeurs uniques	Supprimer colonne après avoir remplacé 'Accident_Id' par 'Num_Acc'
'jour' : dtype('int64')	Faible variation des valeurs	Colonne à supprimer
'mois' : dtype('int64')	-	-
'an' : dtype('int64')	-	Supprimer colonne après avoir calculé l'âge des usagers ('an' - 'an_nais')
'hrmn' : dtype('O')	2 formats horaires existant : HHMM et HH :MM	Convertir HHMM en HH :MM
'lum' : dtype('int64')	Pas de définition pour -1	Dichotomiser toutes les lignes, en excluant celles contenant -1
'dep' : dtype('O')	Trop de valeurs uniques	Colonne à supprimer
'com' : dtype('O')	Trop de valeurs uniques	Colonne à supprimer
'agg' : dtype('int64')	-	-
'int' : dtype('int64')	-	-
'atm' : dtype('int64')	-	Dichotomiser toutes les lignes, en excluant celles contenant -1 + Regrouper des valeurs de météo extrême
'col' : dtype('int64')	-	Dichotomiser toutes les lignes, en excluant celles contenant -1
'adr' : dtype('O')	Trop de valeurs uniques	Colonne à supprimer
'lat' : dtype('O')	Trop de valeurs uniques	Colonne à supprimer
'long' : dtype('O')	Trop de valeurs uniques	Colonne à supprimer

### 3.1.2 Structure pour 'lieux' :

Colonne	Problématique	Modification/Suppression
'Num_Acc' : dtype('int64')	Trop de valeurs uniques	Colonne à supprimer
'catr' : dtype('int64')	-	Dichotomiser toutes les lignes, en excluant celles contenant 9
'voie' : dtype('O')	Trop de valeurs uniques	Colonne à supprimer
'v1' : dtype('int64')	Informations non pertinentes + Trop de valeurs nulles	Colonne à supprimer
'v2' : dtype('O')	Trop de valeurs uniques	Colonne à supprimer
'circ' : dtype('int64')	Pas de définition pour 0 + Valeurs manquantes existantes	Dichotomiser toutes les lignes, en excluant celles contenant -1
'nbv' : dtype('O')	-	Dichotomiser avec regroupement des valeurs supérieures à 5
'vosp' : dtype('int64')	Valeurs manquantes existantes	Dichotomiser toutes les lignes, en excluant celles contenant -1 et NaN
'prof' : dtype('int64')	Valeurs manquantes existantes	Dichotomiser toutes les lignes, en excluant celles contenant -1 et NaN
'pr' : dtype('O')	Trop de valeurs uniques + Valeurs manquantes existantes	Colonne à supprimer
'pr1' : dtype('O')	Trop de valeurs uniques + Valeurs manquantes existantes	Colonne à supprimer
'plan' : dtype('int64')	Valeurs manquantes existantes	Dichotomiser toutes les lignes, en excluant celles contenant -1 et NaN
'lartpc' : dtype('O')	Trop de valeurs uniques	Colonne à supprimer
'larrout' : dtype('O')	Trop de valeurs uniques	Colonne à supprimer
'surf' : dtype('int64')	Valeurs manquantes existantes	Dichotomiser toutes les lignes en excluant celles content -1 et NaN
'infra' : dtype('int64')	-	Dichotomiser avec regroupement des valeurs supérieures à 2 et exclusion des lignes content -1 et 0
'situ' : dtype('int64')	-	Dichotomiser toutes les lignes, en excluant celles contenant -1
'vma' : dtype('int64')	Valeurs parfois aberrantes de : -1 à 901 kmh	Dichotomiser toutes les lignes avec une vitesse comprise entre 0 et 130 km/h

### 3.1.3 Structure pour 'usagers' :

Colonne	Problématique	Modification/Suppression
'Num_Acc' : dtype('int64')	Trop de valeurs uniques	Colonne à supprimer
'id_usager' : dtype('O')	Trop de valeurs uniques	Colonne à supprimer
'id_véhicule' : dtype('O')	Trop de valeurs uniques	Colonne à supprimer
'num_veh' : dtype('O')	Trop de valeurs uniques	Colonne à supprimer
'place' : dtype('int64')	Valeurs manquantes existantes	Dichotomiser toutes les lignes, en excluant celles contenant -1
'catu' : dtype('int64')	Valeurs manquantes existantes	Dichotomiser toutes les lignes, en excluant celles contenant -1
'grav' : dtype('int64')	Pas de définition pour -1	Dichotomiser toutes les lignes, en excluant celles contenant -1
'sexe' : dtype('int64')	-	Dichotomiser toutes les lignes
'an_nais' : dtype('float64')	-	Supprimer colonne après avoir calculé l'âge des usagers ('an' - 'an_nais')
'trajet' : dtype('int64')	-	Dichotomiser toutes les lignes, en excluant celles contenant -1
'secu1' : dtype('int64')	-	-
'secu2' : dtype('int64')	-	-
'secu3' : dtype('int64')	-	-
'locp' : dtype('int64')	-	Dichotomiser toutes les lignes, en excluant celles contenant -1
'actp' : dtype('O')	Valeurs manquantes existantes	Dichotomiser toutes les lignes, en excluant celles contenant -1, 0 et B
'etatp' : dtype('int64')	-	Dichotomiser toutes les lignes, en excluant celles contenant -1

### 3.1.4 Structure pour 'véhicules' :

Colonne	Problématique	Modification/Suppression
'Num_Acc' : dtype('int64')	Trop de valeurs uniques	Colonne à supprimer
'id_véhicule' : dtype('O')	Trop de valeurs uniques	Colonne à supprimer
'num_veh' : dtype('O')	Trop de valeurs uniques	Colonne à supprimer
'senc' : dtype('int64')	-	Dichotomiser toutes les lignes, en excluant celles contenant -1
'catv' : dtype('int64')	Pas de définition de -1	Dichotomiser toutes les lignes, en excluant celles contenant -1 et 0
'obs' : dtype('int64')	-	Dichotomiser toutes les lignes, en excluant celles contenant -1 et 0
'obsm' : dtype('int64')	-	Dichotomiser toutes les lignes, en excluant celles contenant -1 et 0
'choc' : dtype('int64')	-	Dichotomiser toutes les lignes, en excluant celles contenant -1 et 0
'manv' : dtype('int64')	-	Dichotomiser toutes les lignes, en excluant celles contenant -1 et 0
'motor' : dtype('int64')	-	Dichotomiser toutes les lignes, en excluant celles contenant -1 et 0
'occute' : dtype('float64')	Trop de valeurs uniques	Colonne à supprimer

## 3.2 Etapes de transformation des dataframes

### 3.2.1 Initialisation de l'environnement

Dans cette section, nous mettons en place un environnement de travail pour le projet. Il a notamment pour but de structurer notre flux de travail, documenter nos actions effectuées, et garantir la reproductibilité des résultats.

Voici plus précisément la description des étapes concrétisées dans notre code :

- Il permet d'importer les bibliothèques nécessaires, définir les chemins de répertoires où seront stockés les différents types de fichiers, mettre en place un système de journalisation des actions effectuées au cours du

traitement. Ce système utilise la fonction 'log\_action' qui affiche les actions et alimente en même temps la liste 'actions' permettant de stocker les opérations réalisées. La liste d'actions est affichée en fin de notebook pour servir à la rédaction du rapport.

- Une liste 'var\_ecartees' est simplement initialisée pour servir à l'enregistrement des variables écartées. Celle-ci sera complétée ultérieurement pour permettre la prise en compte des noms et motifs d'écartement de ces variables.
- Dans ce même code, deux fichiers de configuration au format JSON sont chargés :
  - le premier (desc\_fic\_raw.json) contient des informations sur les fichiers de données brutes, comme leurs noms, les séparateurs utilisés, et les périodes concernées. La description des fichiers a été motivée par les disparités des fichiers et elle a permis charger facilement toutes les données.
  - le second (desc\_vars.json) décrit les variables du jeu de données, incluant leurs libellés et les correspondances pour les modalités, ce qui permettra des affichages avec les libellés correspondants aux nom de variables et codes de modalités bien plus explicites.
- Un dictionnaire 'dfrub' est initialisé en vue d'y stocker des dataframes dans les prochaines étapes du traitement.

#### Code Python

```

1 import pandas as pd
2 import json
3
4 rep_src  = "data/raw/"          # Fichiers téléchargés avant traitement
5 rep_inter = "data/inter/"        # Des fichiers intermédiaires sont utilisés
6 rep_dst   = "data/processed/"    # Fichiers utilisés par la modélisation
7
8
9 # Recensement des actions réalisées
10 actions = []                  # initialisation d'une liste des actions réalisées
11 def log_action(str):           # fonction pour stocker toutes les actions réalisées
12     dans la liste 'actions'
13     print (f" - {str}")
14     actions.append(str)
15
16 # Initialisation d'une liste des variables écartées
17 var_ecartees = []
18
19 # Lecture du fichier JSON de description des fichiers
20 with open("./desc_fic_raw.json", 'r', encoding='utf-8') as fichier:
21     des_fic_raw = json.load(fichier)
22
23 # Lecture du fichier JSON de description des variables
24 with open("./desc_vars.json", 'r', encoding='utf-8') as fichier:
25     desc_vars = json.load(fichier)
26
27 # Initialisation d'un dictionnaire pour la future concaténation des dataframes
28 dfrub = {}

```

A la base, les données sont réparties en 4 rubriques qui ont été volontairement réduites sur 4 années successives (2019-2022).

Pour mettre en oeuvre leur prétraitement, nous utiliserons le fichier desc\_fic\_raw.json qui contient les informations couvrant divers aspects des accidents de la route, des véhicules impliqués et des personnes concernées. Ci-après quelques éléments correspondants :

- Le nom du fichier ;
- Le séparateur ;
- La phase : jusqu'à 2018 ou à partir de 2019 ;
- Les conversions de types dans dtypes à réaliser lors du chargement.

Le code de la cellule suivante effectue le traitement et la consolidation de données à partir de plusieurs fichiers CSV, organisés par rubriques. Voici comment il fonctionne :

- Il commence à parcourir chaque rubrique définie dans un dictionnaire appelé "des\_fic\_raw".
- Pour chaque rubrique, le code initialise un compteur pour suivre le nombre total d'observations. Il prépare également une liste vide pour stocker les DataFrames pandas qui seront créés pour chaque fichier annuel.
- Ensuite, le script traite chaque fichier annuel associé à la rubrique en cours. Il ne traite que les fichiers marqués comme étant en "phase 2". Pour chaque fichier éligible, il effectue les opérations suivantes :
  - Lecture du fichier CSV en utilisant pandas, en spécifiant les paramètres appropriés tels que le séparateur, l'encodage et les types de données.
  - Comptage du nombre d'observations dans le fichier.
  - Si nécessaire, renommage des colonnes du DataFrame selon les spécifications fournies.
  - Ajout du DataFrame à la liste préparée précédemment.
- Une fois tous les fichiers d'une rubrique traités, le code concatène tous les DataFrames de la liste en un seul grand DataFrame. Ce DataFrame consolidé subit ensuite un nettoyage : les valeurs "-1" (avec un espace) sont remplacées par "-1" pour corriger un problème de codage des valeurs manquantes.
- Le script affiche ensuite des statistiques sur le traitement effectué, notamment le nombre de DataFrames traités, le nombre total d'observations et le nombre de colonnes dans le DataFrame final.
- Enfin, le DataFrame consolidé et nettoyé est sauvegardé dans un nouveau fichier CSV, avec le nom de la rubrique.
- Le processus se répète pour chaque rubrique, créant ainsi un ensemble de fichiers CSV consolidés, chacun correspondant à une rubrique spécifique.
- Pour terminer, le code crée des DataFrames distincts pour chaque rubrique principale : usagers, véhicules, lieux et caractéristiques.

#### Code Python

```

1 # Parcours de chaque rubrique dans le dictionnaire 'des_fic_raw'
2 for rubrique, description_rub in des_fic_raw.items():    # Pour chaque rubrique
3     # Initialisation du compteur d'observations pour la rubrique
4     nb_obs = 0    # nombre total d'observations
5     print ()
6     print("Rubrique : ", rubrique)
7     # Préparation de la liste pour stocker les DataFrames
8     dfl = []
9     # Récupération des types de données pour la rubrique
10    dtype = description_rub.get("dtypes")
11
12    # Traitement de chaque fichier annuel pour la rubrique
13    for fichier_origine in description_rub["fichiers"] :   # Pour chaque fichier annuel
14        # Vérification si le fichier est en phase 2
15        if fichier_origine.get("phase") == "2":
16            nom_fichier = fichier_origine["nom"]
17            # Lecture du fichier CSV
18            df = pd.read_csv(rep_src + nom_fichier,
19                            sep=fichier_origine["sep"],
20                            dtype=dtype,
21                            encoding="latin_1",
22                            index_col=False,
23                            quotechar="\"",
24                            low_memory=False)
25            # Mise à jour du nombre total d'observations
26            nb_obs += df.shape[0]
27            # Renommage des colonnes si nécessaire
28            if fichier_origine.get("rename_cols") is not None:
29                df = df.rename(columns=fichier_origine.get("rename_cols"))
30                print(" - rename ", fichier_origine.get("rename_cols"))
31            # Affichage des informations sur le fichier traité
32            print(nom_fichier, df.shape)
33            # Ajout du DataFrame à la liste
34            dfl.append(df)
35
36    # Concaténation de tous les DataFrames annuels (2019->2022) en un seul

```

```

37 dfrub[rubrique] = pd.concat(dfl)
38
39 # Remplacement des " -1" par "-1" pour les valeurs manquantes
40 dfrub[rubrique] = dfrub[rubrique].replace("-1", "-1")
41
42 # Affichage des statistiques de traitement
43 print("Nombre de DataFrames : ", len(dfl))
44 print("Nombre d'observations : ", nb_obs, dfrub[rubrique].shape[0])
45 print("Colonnes du DataFrame : ", dfrub[rubrique].shape[1])
46
47 # Sauvegarde du DataFrame consolidé et nettoyé
48 dfrub[rubrique].to_csv(rep_dst + rubrique + ".csv", sep='\t', index=False)
49
50 # Création d'un DataFrame propre à chaque rubrique principale
51 dfc = dfrub["caracteristiques"]
52 dfl = dfrub["lieux"]
53 dfu = dfrub["usagers"]
54 dfv = dfrub["vehicules"]

```

Ce script permet donc d'automatiser le processus de consolidation et de nettoyage initial des données provenant des multiples fichiers annuels, en les regroupant par catégorie et en effectuant quelques corrections basiques.

### 3.2.2 Enregistrement des 4 rubriques de 2019 à 2022

Après avoir nettoyé et uniformisé les données brutes, nous enregistrons les jeux de données de chaque rubrique — caractéristiques, lieux, usagers et véhicules — couvrant la période de 2019 à 2022. Ces enregistrements servent de fichiers intermédiaires qui pourront être utilisés pour des explorations et des analyses plus approfondies, tout en permettant de valider les étapes de prétraitement effectuées.

En sauvegardant les données à ce stade, nous nous assurons que les ajustements et transformations appliqués sont bien capturés avant de procéder à des étapes plus avancées, telles que la jointure des jeux de données. Cela facilite également la détection et la correction d'éventuelles erreurs ou incohérences rencontrées lors du prétraitement, en permettant un retour en arrière sans devoir répéter les étapes initiales de nettoyage.

#### Code Python

```

1 # Sauvegarde des jeux de données nettoyés et standardisés de 2019 à 2022 pour chaque
   rubrique.
2 # Ces fichiers intermédiaires sont enregistrés au format CSV pour un usage ultérieur.
3 dfc.to_csv(rep_inter+'caracteristiques_raw_4.csv', sep = '\t')
4 dfl.to_csv(rep_inter+'lieux_raw_4.csv', sep = '\t')
5 dfu.to_csv(rep_inter+'usagers_raw_4.csv', sep = '\t')
6 dfv.to_csv(rep_inter+'vehicules_raw_4.csv', sep = '\t')

```

### 3.2.3 Jointure des 4 rubriques

Nous cherchons à prédire la gravité des accidents pour les personnes en fonction des circonstances des accidents, la gravité est dans la variable grav de la rubrique usagers, c'est notre cible, la rubrique usagers contient déjà quelques informations, nous relions alors les informations des autres rubriques à la rubrique usagers. Cette étape implique la jointure des données sur la base de l'identifiant unique d'accident (Num\_Acc). Les jointures ne doivent pas perdre des données ou introduire des erreurs dues à des correspondances incorrectes. Nous les vérifions à chaque fois en affichant les nombres de colonnes (variables) et de lignes (observations).

Notre code réalise ainsi plusieurs opérations menant à la jointure des différents DataFrames créés. Quelques remarques préalables à ce sujet :

- Les jointures sont toutes faites avec le champ Num\_Acc ;
- Le type de Num\_Acc est forcé à "int" lors de la lecture par read\_csv() ;
- La dernière jointure sur les véhicules est faite avec, en plus, les champs num\_veh et id\_vehicule ;
- Les jointures sont "à gauche" ("left") pour conserver le nombre d'usagers ;
- Les nombres d'observations et de variables affichés avant et après chaque jointure permettent de vérifier les jointures ;

— Il y a 494 182 usagers avant les jointures et le DataFrame final 'df' résultant a ce même nombre d'observations.

Après les jointures, le code utilise la méthode info() pour afficher des informations détaillées sur le DataFrame, y compris les types de données et le nombre de valeurs non nulles pour chaque colonne. Enfin, il libère de la mémoire en mettant à None les DataFrames qui ne seront plus utilisés (dfc, dfl, dfu, dfv, dfrub).

Tout au long du processus, le code utilise la fonction log\_action() pour enregistrer les étapes principales du processus de jointure.

#### Code Python

```
1 ######
2 # Jointures des 4 rubriques
3 #####
4
5 print("")
6 log_action("Jointure usagers <--- caractéristiques")
7 print()
8
9 # Affichage des dimensions des DataFrames avant la jointure
10 print("Taille usagers : ", dfu.shape)
11 print("Taille caractéristiques : ", dfc.shape)
12
13 # Première jointure : usagers et caractéristiques
14 df = pd.merge(dfu, dfc, on="Num_Acc", how = "left") # jointure à gauche pour conserver
    toutes les lignes de 'usagers'
15 print("Tailles df résultant : ", df.shape)
16 print("")
17
18 log_action("Jointure (usagers et caractéristiques) <--- lieux")
19 print()
20
21 # Affichage des dimensions avant la deuxième jointure
22 print("Taille DataFrame : ", df.shape)
23 print("Taille lieux : ", dfl.shape)
24
25 # Deuxième jointure : ajout des données de lieux
26 df = pd.merge(df, dfl, on="Num_Acc", how = "left")
27 print("Tailles df résultant : ", df.shape)
28 print("")
29
30 log_action("Jointure (usagers, caractéristiques et lieux) <--- véhicules")
31 print()
32
33 # Affichage des dimensions avant la troisième jointure
34 print("Taille DataFrame : ", df.shape)
35 print("Taille véhicules : ", dfv.shape)
36
37 # Troisième jointure : ajout des données de véhicules
38 df = pd.merge(df, dfv, on=["Num_Acc", "id_vehicule", "num_veh"], how = "left") #
    jointure sur plusieurs colonnes
39 print("Tailles df résultant : ", df.shape)
40 print()
41
42 # Affichage des informations sur le DataFrame final
43 print("Colonnes résultantes : ", df.columns)
44 print(df.info(max_cols=1000, show_counts=True))
45
46 # Libération de la mémoire (on supprime les DataFrames des 4 rubriques qui ne seront
    plus utilisés)
47 dfc = None
48 dfl = None
49 dfu = None
```

```
50 dfv = None  
51 dfrub = None
```

### 3.2.4 Suppression d'observations

Lors de la préparation des données pour l'analyse et la modélisation, il est essentiel de s'assurer que les observations incluses soient complètes et cohérentes. La variable 'grav' représente la gravité de l'accident, qui est une information clé pour toute analyse visant à comprendre ou prédire les facteurs influençant la gravité des accidents de la route.

Cependant, certaines observations contiennent une valeur de 'grav' égale à -1, indiquant que la gravité est inconnue ou non renseignée. Ces observations ne donnent pas d'information et risquent de biaiser nos modèles.

Pour cette raison, nous décidons de supprimer toutes les observations pour lesquelles la gravité est inconnue ('grav' = -1). Cette étape garantit que le jeu de données final est composé uniquement de cas où la gravité de l'accident est clairement définie, ce qui améliore la qualité des données pour les analyses futures.

#### Code Python

```
1 # Suppression des observations dont la gravité de l'accident ('grav') est inconnue  
# (valeur '-1')  
2  
3 # Enregistrement du nombre d'observations avant le nettoyage  
4 nb_avant = df.shape[0]  
5 print(f"Nombre d'observations avant suppression {nb_avant}")  
6  
7 # Filtrage du DataFrame pour exclure les lignes où la gravité est codée '-1' (inconnue)  
8 df1 = df.loc[df.grav != '-1', :]  
9  
10 # Calcul et affichage du nombre d'observations supprimées  
11 print(f"Nombre d'observations supprimées {nb_avant - df.shape[0]}")  
12  
13 # Affichage du nombre d'observations restantes après le nettoyage  
14 print(f"Nombre d'observations après suppression {df.shape[0]}")  
15  
16 # Enregistrement de l'action de nettoyage dans le journal  
17 log_action(f"Suppression de {nb_avant - df.shape[0]} observations dont la gravité est  
inconnue (codée -1)")
```

### 3.2.5 Crédation de variables

La création de nouvelles variables est une étape clé du prétraitement des données, car elle permet de capturer des informations supplémentaires qui ne sont pas directement présentes dans les variables brutes. En générant de nouvelles variables à partir des données existantes, nous pouvons identifier des motifs et des relations qui pourraient être déterminants pour la prédiction de la gravité des accidents de la route. Intuitivement, nous pensons que le jour de la semaine, les jours fériés et l'âge des usagers ont une influence sur les conséquences des accidents.

Dans cette section, nous créons alors trois nouvelles variables qui pourraient apporter des explications supplémentaires de la gravité des accidents :

- **Jour de la semaine (jsem)** : Cette variable est créée pour identifier le jour de la semaine (lundi à dimanche) où l'accident s'est produit. Cette information peut être utile pour comprendre les tendances hebdomadaires, telles que l'augmentation des accidents le week-end ou les jours de semaine chargés.
- **Jour férié** : Les jours fériés peuvent avoir un impact significatif sur le trafic et le comportement des conducteurs, ce qui pourrait influencer la gravité des accidents. Cette variable binaire indique si l'accident s'est produit un jour férié ou non (incluant les dimanches et les jours de fête).
- **Âge des usagers (age)** : Calculée comme la différence entre l'année de l'accident et l'année de naissance de l'usager, cette variable permet d'analyser l'impact de l'âge sur la gravité des accidents. L'âge des conducteurs, par exemple, peut être un facteur très important, les jeunes conducteurs et les conducteurs plus âgés pouvant avoir des comportements et des risques différents. Lorsque l'année de naissance est inconnue, l'âge est codé -1. Cette variable sera dichotomisée ultérieurement.

Ces variables créées visent à enrichir le jeu de données et à fournir des attributs supplémentaires qui peuvent améliorer la performance des modèles prédictifs en capturant des dimensions supplémentaires de la dynamique des accidents de la route.

#### Code Python

```

1 ######
2 # Création de la variable 'jsem' pour représenter l'influence du jour de la semaine sur
3 # - jsem : jour de la semaine : lundi : 1 , ... , 7 : dimanche
4 #####
5
6 print(" - Inclusion du jour de la semaine dans le fichier")
7
8 # Création d'un DataFrame temporaire avec les colonnes de date
9 df_date = df[["an", "mois", "jour"]]
10 df_date = df_date.rename({"an": "year", "mois": "month", "jour": "day"}, axis=1)
11
12 # Conversion en timestamp
13 df_date["ts"] = pd.to_datetime(df_date)
14
15 # Calcul du jour de la semaine (1-7 au lieu de 0-6)
16 df_date["jsem"] = df_date.ts.apply(lambda x: x.weekday()+1)
17
18 # Ajout de la colonne jsem au DataFrame principal
19 df["jsem"] = df_date.jsem
20 df_date = None # Libération de la mémoire
21
22 print ("Jours de la semaine : ")
23 print (df.jsem.value_counts().sort_index())
24 log_action (f"Création de la variable jsem : jour de la semaine")
25
26 #####
27 # Création de la variable 'ferie' pour représenter l'influence des jours fériés sur la
28 # - gravité de l'accident :
29 # - ferie : 0 jour ouvré, 1 : jour ferié
30 # Les jours fériés sont les dimanches et les jours de fête.
31 #####
32 df["ferie"] = False
33
34 # Marquage des jours fériés
35
36 # Les dimanches
37 df.loc[df.jsem==7,"ferie"] = True
38
39 # Les 1er janvier
40 df.loc[(df.mois==1) & (df.jour==1), 'ferie'] = True
41
42 # Les 1er mai
43 df.loc[(df.mois==5) & (df.jour==1), "ferie"] = True
44
45 # Les 8 mai
46 df.loc[(df.mois==5) & (df.jour==8), "ferie"] = True
47
48 # Les 14 juillet
49 df.loc[(df.mois==7) & (df.jour==14), "ferie"] = True
50
51 # Les 15 août
52 df.loc[(df.mois==7) & (df.jour==14), "ferie"] = True
53
54 # Les 1er novembre

```

```

55 df.loc[(df.mois==11) & (df.jour==1), "ferie"] = True
56
57 # Les 11 novembre
58 df.loc[(df.mois==11) & (df.jour==11), "ferie"] = True
59
60 # Les 25 décembre
61 df.loc[(df.mois==12) & (df.jour==25), "ferie"] = True
62
63 # Dimanche de pâques, date variant chaque année
64 df.loc[(df.an == 2019) & (df.mois==4) & (df.jour==21), "ferie"] = True
65 df.loc[(df.an == 2020) & (df.mois==4) & (df.jour==12), "ferie"] = True
66 df.loc[(df.an == 2021) & (df.mois==4) & (df.jour==4), "ferie"] = True
67 df.loc[(df.an == 2022) & (df.mois==4) & (df.jour==17), "ferie"] = True
68
69 # Lundi de pentecôte, date variant chaque année
70 df.loc[(df.an == 2019) & (df.mois==6) & (df.jour==10), "ferie"] = True
71 df.loc[(df.an == 2020) & (df.mois==6) & (df.jour==1), "ferie"] = True
72 df.loc[(df.an == 2021) & (df.mois==5) & (df.jour==24), "ferie"] = True
73 df.loc[(df.an == 2022) & (df.mois==6) & (df.jour==6), "ferie"] = True
74
75 # Jeudi de l'ascension, date variant chaque année
76 df.loc[(df.an == 2019) & (df.mois==5) & (df.jour==30), "ferie"] = True
77 df.loc[(df.an == 2020) & (df.mois==5) & (df.jour==21), "ferie"] = True
78 df.loc[(df.an == 2021) & (df.mois==5) & (df.jour==13), "ferie"] = True
79 df.loc[(df.an == 2022) & (df.mois==5) & (df.jour==26), "ferie"] = True
80
81 # Saint Étienne en alsace Moselle, le 26 décembre
82 df.loc[df.dep.isin([57, 67, 68]) & (df.mois==12) & (df.jour==26), "ferie"] = True
83
84 # Abolition de l'esclavage en Guadeloupe
85 df.loc[(df.dep == 971) & (df.mois==5) & (df.jour==27), "ferie"] = True
86
87 # Abolition de l'esclavage en Martinique
88 df.loc[(df.dep == 972) & (df.mois==5) & (df.jour==22), "ferie"] = True
89
90 # Abolition de l'esclavage en Guyane
91 df.loc[(df.dep == 973) & (df.mois==6) & (df.jour==10), "ferie"] = True
92
93 # Abolition de l'esclavage à la Réunion
94 df.loc[(df.dep == 974) & (df.mois==12) & (df.jour==20), "ferie"] = True
95
96 # Abolition de l'esclavage à Mayotte
97 df.loc[(df.dep == 976) & (df.mois==4) & (df.jour==27), "ferie"] = True
98
99 # Abolition de l'esclavage à Saint-Barthélémy
100 df.loc[(df.dep == 977) & (df.mois==10) & (df.jour==9), "ferie"] = True
101
102 # Abolition de l'esclavage à Saint-Martin
103 df.loc[(df.dep == 978) & (df.mois==5) & (df.jour==28), "ferie"] = True
104
105 print ("Jours ouvrés et fériés")
106 print (df.ferie.value_counts())
107 log_action (f"Création de la variable ferie : jour férié, dimanches et autres fêtes")
108
109 #####
110 #####
111 # Création de la variable 'age' pour représenter l'influence de l'âge sur la gravité des
     accidents
112 # Elle est égale à l'année de l'accident moins l'année de naissance à un an près.
113 # L'année de naissance n'est pas toujours connue, elle est alors codée -1
114 #####

```

```

115
116 print (df.an_nais.isna().sum())
117
118 # Calcul de l'âge
119 ag = df.an.astype("int") - df.an_nais.fillna(2030).astype ("int") # valeur négative
    pour l'âge si date de naissance inconnue
120 ag.loc[ag<0] = -1 # âge inconnu codé comme -1
121 df["age"] = ag
122 ag = None # Libération de la mémoire
123
124 log_action (f"Création de la variable age : différence entre l'année de l'accident et
    l'année de naissance")

```

### 3.2.6 Dichotomisation / catégorisation

Dans le cadre de la préparation des données pour la modélisation prédictive, certaines variables sont transformées par dichotomisation (transformation en variables binaires) ou par catégorisation pour mieux capturer des informations pertinentes et améliorer la performance des modèles d'apprentissage.

#### Dichotomisation des variables :

La dichotomisation consiste à convertir une variable en une variable binaire (0 ou 1). Cette méthode est utilisée lorsque nous souhaitons simplifier une variable en la réduisant à deux catégories distinctes. Par exemple, la variable 'lum' (luminosité) pourrait être transformée en une variable binaire pour distinguer les conditions de jour (1) des conditions de nuit (0). Les valeurs non définies (comme -1) sont généralement exclues pour éviter les biais.

#### Catégorisation des variables :

La catégorisation consiste à diviser une variable en plusieurs catégories distinctes. Cette méthode est utilisée pour regrouper des valeurs en classes significatives, gérer les valeurs aberrantes ou simplifier des variables avec de nombreuses modalités. Par exemple, pour la variable 'vma' (vitesse maximale autorisée), qui peut contenir des valeurs allant de -1 à 901 km/h, il est possible de créer plusieurs catégories comme "0-50 km/h", "51-90 km/h", et "91-130 km/h". Cette transformation permet de structurer les données de manière plus interprétable et d'améliorer la robustesse des modèles.

#### Regroupement de modalités et gestion des valeurs manquantes :

Les variables telles que circ (circulation) ou prof (profil de la route) contiennent des valeurs non définies ou manquantes. Celles-ci sont regroupées ou exclues pour s'assurer que seules des données valides et pertinentes sont utilisées dans les analyses et la modélisation. Par exemple, la variable circ est dichotomisée en excluant les valeurs -1 et autres modalités non pertinentes, ce qui permet de rendre les données plus cohérentes.

En appliquant ces techniques, nous simplifions la structure des données, ce qui peut améliorer les performances des modèles prédictifs et faciliter l'interprétation des résultats.

Voici un résumé des principales transformations effectuées :

##### 1. Variables temporelles :

- 'jsem' (jour de la semaine) : dichotomisée en 7 variables binaires
- 'hrmn' (heure) : catégorisée en 5 périodes (matin, midi, après-midi, soir, nuit)
- 'mois' : transformée en 12 variables binaires

##### 2. Variables démographiques :

- 'age' : catégorisée en 4 groupes (enfant, jeune, adulte, 3ème âge)
- 'sexe' : transformée en variables binaires (homme/femme)

##### 3. Variables liées à la sécurité :

- 'secu1', 'secu2', 'secu3' : combinées en 6 variables binaires représentant différents équipements de sécurité

##### 4. Variables liées à l'environnement :

- 'surf' (état de la surface) : catégorisée en 4 types (normale, mouillée, glissante, autre)
- 'atm' (conditions atmosphériques) : dichotomisée en excluant les valeurs non pertinentes
- 'lum' (luminosité) : dichotomisée en excluant les valeurs non définies

##### 5. Variables liées au véhicule et à la route :

- 'catv' (catégorie de véhicule) : transformée en plusieurs variables binaires en excluant les valeurs non pertinentes
- 'nbv' (nombre de voies) : regroupée en 5 catégories (1 à 4 voies, et 5+)

— 'vma' (vitesse maximale autorisée) : catégorisée en groupes de vitesse pertinents (30km/h et moins, 40-50km/h, 60-70km/h, 80-90km/h, 100km/h et plus)

#### 6. Variable cible :

— 'grav' (gravité) : dichotomisée en 'grave' et 'non grave'

Autres variables dichotomisées : 'choc', 'manv', 'obs', 'obsm', 'catu', 'trajet', 'motor', entre autres.

Note : Pour toutes ces transformations, nous avons veillé à exclure les valeurs non pertinentes ou inconnues (souvent codées comme -1) afin de ne pas biaiser notre analyse.

#### Code Python

```
1 ######
2 # Fonction de dichotomisation adaptée
3 #####
4
5 def dichotomisation (df, column, var_ecartees, desc_vars=None, dummies=None,
6     mod_ecartees = None):
7     """
8     Cette fonction fait la dichotomisation des seules modalités de la
9     liste fournie par dummies. Elle permet de ne pas dichotomiser les
10    modalités : "non renseigné", "Autre", "Non applicable", ...
11    Elle utilise si possible les infos de desc_vars pour nommer les colonnes.
12    et elle complète desc_vars.
13    """
14    # Récupération des descriptions de colonnes si disponibles
15    try :
16        desc_vars = desc_vars.get("columns")
17        col_desc = {}
18        for c in desc_vars :
19            if c.get("name") == column:
20                col_desc = c
21                break
22    except :
23        col_desc = {}
24
25    # Détermination des modalités à dichotomiser
26    if dummies is None:
27        dum = list(df[column].unique())
28    else:
29        dum = dummies
30
31    # Exclusion des modalités à écarter
32    if mod_ecartees is not None:
33        dum = [x for x in dum if x not in mod_ecartees]
34    print()
35
36    # Traitement de chaque modalité
37    for c in dum: # c : modalité
38
39        # Création du nom de la nouvelle colonne
40        new_col_name = column + "_" + str(c)
41
42        # Création de la nouvelle variable dichotomique
43        df[new_col_name] = df[column] == c
44
45        # Recherche d'une description existante pour la nouvelle colonne
46        desc_new_col = None
47        for ic in range(len(desc_vars)):
48            if desc_vars[ic].get("name") == new_col_name:
49                desc_new_col = desc_vars[ic]
50                break
51
52        # Si aucune description n'existe, en créer une nouvelle
```

```

52     if desc_new_col is None:
53         desc_new_col = {}
54     desc_new_col["name"] = new_col_name
55     desc_new_col["dtype"] = "bool"
56     values = col_desc.get("values")
57     if values is not None and values.get(c) is not None:
58         desc_new_col["label"] = col_desc.get("label") + " : " + values.get(c)
59     else :
60         desc_new_col["label"] = col_desc.get("label")
61     desc_vars.append(desc_new_col)
62
63 # Ajout de la colonne d'origine à la liste des variables écartées
64 var_ecartees.append((column, "Dichotomisation"))
65
66 return

```

### Code Python

```

1 ##########
2 # Dichotomisation des champs secu 1 à 3
3 #########
4 # L'usager peut être protégé par plusieurs équipements, il y a trois variables
5 # pour décrire ces équipements, il y a 6 types d'équipement, il faut alors
6 # créer six modalités.
7
8 df["secu_ceinture"] = False
9 df["secu_casque"] = False
10 df["secu_dispenfant"] = False
11 df["secu_gilet"] = False
12 df["secu_airbag23RM"] = False
13 df["secu_gants"] = False
14
15 df.loc[df.sec1 == '1', "secu_ceinture"] = True
16 df.loc[df.sec2 == '1', "secu_ceinture"] = True
17 df.loc[df.sec3 == '1', "secu_ceinture"] = True
18
19 df.loc[df.sec1 == '2', "secu_casque"] = True
20 df.loc[df.sec2 == '2', "secu_casque"] = True
21 df.loc[df.sec3 == '2', "secu_casque"] = True
22
23 df.loc[df.sec1 == '3', "secu_dispenfant"] = True
24 df.loc[df.sec2 == '3', "secu_dispenfant"] = True
25 df.loc[df.sec3 == '3', "secu_dispenfant"] = True
26
27 df.loc[df.sec1 == '4', "secu_gilet"] = True
28 df.loc[df.sec2 == '4', "secu_gilet"] = True
29 df.loc[df.sec3 == '4', "secu_gilet"] = True
30
31 df.loc[df.sec1 == '5', "secu_airbag23RM"] = True
32 df.loc[df.sec2 == '5', "secu_airbag23RM"] = True
33 df.loc[df.sec3 == '5', "secu_airbag23RM"] = True
34 df.loc[df.sec1 == '7', "secu_airbag23RM"] = True
35 df.loc[df.sec2 == '7', "secu_airbag23RM"] = True
36 df.loc[df.sec3 == '7', "secu_airbag23RM"] = True
37
38 df.loc[df.sec1 == '6', "secu_gants"] = True
39 df.loc[df.sec2 == '6', "secu_gants"] = True
40 df.loc[df.sec3 == '6', "secu_gants"] = True
41 df.loc[df.sec1 == '7', "secu_gants"] = True
42 df.loc[df.sec2 == '7', "secu_gants"] = True

```

```

43 df.loc[df.secu3 == '7', "secu_gants"] = True
44
45 var_ecartees.append(("secu1", "Dichotomisation"))
46 var_ecartees.append(("secu2", "Dichotomisation"))
47 var_ecartees.append(("secu3", "Dichotomisation"))
48 log_action(f"Dichotomisation des champs secu1, secu2 et secu3")
49
50 ######
51 # Dichotomisation de l'âge
52 #####
53 df["age_enfant"] = (df.age>=0) & (df.age<= 15)
54 df["age_jeune"] = (df.age>15) & (df.age<= 25)
55 df["age_adulte"] = (df.age>25) & (df.age<= 64)
56 df["age_3age"] = (df.age>64)
57
58 var_ecartees.append(("age", "Dichotomisation"))
59 log_action(f"Dichotomisation de l'âge")
60
61 #####
62 # Dichotomisation de l'heure
63 #####
64 df["hr_matin"] = (df.hrmn>="0600") & (df.hrmn<"1200")
65 df["hr_midi"] = (df.hrmn>="1200") & (df.hrmn<"1400")
66 df["hr_am"] = (df.hrmn>="1400") & (df.hrmn<"1800")
67 df["hr_soir"] = (df.hrmn>="1800") & (df.hrmn<"2100")
68 df["hr_nuit"] = (df.hrmn>="2100") | (df.hrmn<"0600")
69
70 var_ecartees.append(("hrmn", "Dichotomisation"))
71 log_action(f"Dichotomisation de l'heure")
72
73 #####
74 # Dichotomisation du sexe
75 #####
76 df["sexe_m"] = df.sex == '1'
77 df["sexe_f"] = df.sex == '2'
78
79 var_ecartees.append(("sexe", "Dichotomisation"))
80 log_action(f"Dichotomisation du sexe")
81
82 #####
83 # Dichotomisation de la gravité
84 #####
85 # Les noms de variables seront plus faciles à retenir que les valeurs 1 à 4
86 # N.B.: les observations dont la gravité est inconnue (codée -1) seront écartées
87 df["grav_grave"] = df.grav.isin(["2", "3"])
88
89 var_ecartees.append(("grav", "Dichotomisation"))
90 log_action(f"Dichotomisation de la gravité")
91
92 #####
93 # Dichotomisation du nombre de voies de circulation avec regroupement
94 #####
95 df["nbv_1"] = df.nbv == '1'
96 df["nbv_2"] = df.nbv == '2'
97 df["nbv_3"] = df.nbv == '3'
98 df["nbv_4"] = df.nbv == '4'
99 df["nbv_plus"] = df.nbv.isin(['5', '6', '7', '8', '9', '10', '11', '12'])
100
101 var_ecartees.append(("nbv", "Dichotomisation"))
102 log_action(f"Dichotomisation du nombre de voies avec regroupement 1 à 4 puis 5 et plus")
103

```

```

104 #####
105 # Dichotomisation de l'état de la surface
106 #####
107 df["surf_norm"] = df.surf == '1'
108 df["surf_mouil"] = df.surf == '2'
109 df["surf_gliss"] = df.surf.isin(['3', '4', '5', '6', '7', '8', '9'])
110 df["surf_autre"] = df.surf == '9'
111
112 var_ecartees.append(("surf", "Dichotomisation"))
113 log_action(f"Dichotomisation du l'état de la surface : sèche, mouillée, glissante (3 à
114 9)")
115 #####
116 # Dichotomisation de la vitesse maximale autorisée
117 #####
118 # La vma (vitesse maximale autorisée) est codée par un flottant.
119 # Il y a des valeurs aberrantes et codées avec des points décimaux
120 # Nous les convertissons en entiers
121 vma_int = df.vma.astype(int)
122 df["vma_30m"] = vma_int.isin([10, 20, 30])
123 df["vma_40"] = vma_int == 40
124 df["vma_50"] = vma_int == 50
125 df["vma_60"] = vma_int == 60
126 df["vma_70"] = vma_int == 70
127 df["vma_80"] = vma_int == 80
128 df["vma_90"] = vma_int == 90
129 df["vma_110"] = vma_int == 110
130 df["vma_130"] = vma_int == 130
131
132 var_ecartees.append(("vma", "Dichotomisation"))
133 log_action(f"Dichotomisation de la vitesse maximale autorisée avec regroupement")
134
135 #####
136 # Dichotomisation en ou hors agglomération
137 #####
138 # modalités 1 et 2, sans valeurs nulles (-1 ou na)
139 df.agg_agg = df.agg == '1'
140
141 var_ecartees.append(("agg", "Dichotomisation"))
142 log_action("Dichotomisation en ou hors agglomération (agg), 1 agglomération, 0 hors
143 agglomoration")
144 #####
145 # Dichotomisations plus simples
146 #####
147
148 # actp : action du piéton
149 dummies = ['1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B']
150 dichotomisation(df, "actp", var_ecartees, desc_vars, dummies = dummies)
151 log_action("Dichotomisation de l'action du piéton (actp), modalité -1 0 et B exclues")
152
153 # atm : Conditions atmosphériques
154 dummies = ['1', '2', '3', '4', '5', '6', '7', '8']
155 dichotomisation(df, "atm", var_ecartees, desc_vars, dummies = dummies)
156 log_action("Dichotomisation des cond. atmosphériques (atm), modalité -1 et 9 exclues")
157
158 # catr : Catégorie de route
159 dummies = ['1', '2', '3', '4', '5', '6', '7']
160 dichotomisation(df, "catr", var_ecartees, desc_vars, dummies = dummies)
161 log_action("Dichotomisation de la catégorie de route (catr), modalité -1 et 9 exclues")
162
```

```

163 # catu : Catégorie d'usager
164 dummies = ['1', '2', '3']
165 dichotomisation(df, "catu", var_ecartees, desc_vars, dummies = dummies)
166 log_action("Dichotomisation de la catégorie d'usager (catu), modalité -1 exclue")
167
168 # catv : Catégorie de véhicule
169 dichotomisation(df, "catv", var_ecartees, desc_vars, dummies = None, mod_ecartees=[0,
    -1])
170 log_action("Dichotomisation de la catégorie de véhicule (catv), modalité -1 et 0
    exclues")
171
172 # choc : Point de choc initial
173 dichotomisation(df, "choc", var_ecartees, desc_vars, dummies = None, mod_ecartees=['0',
    '-1'])
174 log_action("Dichotomisation du point de choc initial (choc), modalité -1 et 0 exclues")
175
176 # circ : Circulation
177 dichotomisation(df, "circ", var_ecartees, desc_vars, dummies = None, mod_ecartees=['-1'])
178 log_action("Dichotomisation du régime de circulation (circ), modalité -1 exclue")
179
180 # col : Type de collision
181 dichotomisation(df, "col", var_ecartees, desc_vars, dummies = None, mod_ecartees=['-1'])
182 log_action("Dichotomisation du type de collision (col), modalité -1 exclue")
183
184 # etatp: Piéton seul
185 dichotomisation(df, "etatp", var_ecartees, desc_vars, dummies = ['1', '2', '3'])
186 log_action("Dichotomisation de (etatp), modalité -1 exclue")
187
188 # infra : Aménagement - infrastructure
189 dichotomisation(df, "infra", var_ecartees, desc_vars, dummies = None,
    mod_ecartees=['-1', '0'])
190 log_action("Dichotomisation de Aménagement - infrastructure (infra), modalités -1 et 0
    exclues")
191
192 # int : Type d'intersection
193 dichotomisation(df, "int", var_ecartees, desc_vars, dummies = None, mod_ecartees=['-1'])
194 log_action("Dichotomisation du type d'intersection (int), modalité -1 exclues")
195
196 # jsem : Jour de la semaine
197 dichotomisation(df, "jsem", var_ecartees, desc_vars, dummies = None, mod_ecartees=None)
198 log_action("Dichotomisation du jour de la semaine (jsem), toutes modalité ")
199
200 # locp : Localisation du piéton
201 dichotomisation(df, "locp", var_ecartees, desc_vars, dummies = None, mod_ecartees =
    ['-1', '0'])
202 log_action("Dichotomisation de la localisation du piéton (locp), modalités -1, 0(non
    renseigné) exclues")
203
204 # lum : Lumière modalité
205 dichotomisation(df, "lum", var_ecartees, desc_vars, dummies = None, mod_ecartees =
    ['-1'])
206 log_action("Dichotomisation des conditions lumineuses (lum), modalité -1 exclue")
207
208 # manv : Manoeuvre
209 dichotomisation(df, "manv", var_ecartees, desc_vars, dummies = None, mod_ecartees =
    ['-1', '0'])
210 log_action("Dichotomisation de la manoeuvre (manv), modalité -1 et 0 exclues")
211
212 # mois : Mois
213 dichotomisation(df, "mois", var_ecartees, desc_vars, dummies = None, mod_ecartees = None)
214 log_action("Dichotomisation du mois (mois), toutes modalités")

```

```

215
216 # motor : Motorisation
217 dichotomisation(df, "motor", var_ecartees, desc_vars, dummies = None, mod_ecartees =
218     ['-1', '0'])
219 log_action("Dichotomisation de la motorisation (motor), modalité -1 et 0 exclues")
220
221 # obs : Obstacle fixe heurté
222 dichotomisation(df, "obs", var_ecartees, desc_vars, dummies = None, mod_ecartees =
223     ['-1', '0'])
224 log_action("Dichotomisation de l'obstacle fixe heurté (obs), modalité -1 et 0 exclues")
225
226 # obsm : Obstacle mobile heurté
227 dichotomisation(df, "obsm", var_ecartees, desc_vars, dummies = None, mod_ecartees =
228     ['-1', '0'])
229 log_action("Dichotomisation de l'obstacle mobile heurté (obsm), modalité -1 et 0
230     exclues")
231
232 # place : Place de l'usager dans le véhicule
233 dichotomisation(df, "place", var_ecartees, desc_vars, dummies = None, mod_ecartees =
234     ['-1'])
235 log_action("Dichotomisation de la place dans le véhicule (place), modalité -1 exclue")
236
237 # plan : Tracé en plan
238 dichotomisation(df, "plan", var_ecartees, desc_vars, dummies = None, mod_ecartees =
239     ['-1'])
240 log_action("Dichotomisation du tracé en plan (plan), modalité -1 exclue")
241
242 # prof : Déclivité
243 dichotomisation(df, "prof", var_ecartees, desc_vars, dummies = None, mod_ecartees =
244     ['-1'])
245 log_action("Dichotomisation de la déclivité (prof), modalité -1 exclue")
246
247 # senc : Sens de circulation
248 dichotomisation(df, "senc", var_ecartees, desc_vars, dummies = None, mod_ecartees =
249     ['-1'])
250 log_action("Dichotomisation du sens de circulation (senc), modalité -1, 0 et 3 exclues")
251
252 # situ : Situation de l'accident
253 dichotomisation(df, "situ", var_ecartees, desc_vars, dummies = None, mod_ecartees =
254     ['-1'])
255 log_action("Dichotomisation de la situation de l'accident (situ), modalité -1 exclue")
256
257 # trajet : Motif du trajet
258 dichotomisation(df, "trajet", var_ecartees, desc_vars, dummies = None, mod_ecartees =
259     ['-1'])
260 log_action("Dichotomisation du motif du trajet (trajet), modalité -1 exclue")
261
262 # vosp : Présence d'une voie réservée
263 dichotomisation(df, "vosp", var_ecartees, desc_vars, dummies = None, mod_ecartees =
264     ['-1'])
265 log_action("Dichotomisation de la présence d'une voie réservée (vosp), modalités -1
266     exclue")

```

### 3.2.7 Suppression de variables

Certaines variables peuvent contenir des informations redondantes, des valeurs manquantes importantes, des valeurs uniques ou des données non pertinentes qui peuvent nuire à la performance et à la robustesse des modèles prédictifs. En éliminant ces variables, nous réduisons la dimensionnalité des données, ce qui simplifie le modèle, améliore sa capacité à généraliser et réduit les risques de surajustement.

Les raisons principales de la suppression de variables incluent :

**Faible Variabilité ou Valeurs Uniques Trop Nombreuses :**

Certaines variables, comme les identifiants (Num\_Acc, id\_vehicule), contiennent des valeurs uniques pour chaque observation, ce qui n'apporte aucune information utile pour l'analyse prédictive. Ces variables sont souvent éliminées car elles ne contribuent pas à la compréhension des relations entre les données.

#### **Valeurs Manquantes ou Erronées :**

Certaines variables contiennent un nombre excessif de valeurs manquantes ou de valeurs aberrantes, rendant leur utilisation impraticable. Par exemple, une variable qui a plus de 50% de valeurs manquantes peut être supprimée car sa reconstruction ou son imputation pourrait introduire des biais.

#### **Redondance avec d'autres Variables :**

Les variables qui sont fortement corrélées avec d'autres variables peuvent être redondantes (exemple, les variables dichotomisées). Dans ce cas, une seule variable représentative est conservée pour éviter des calculs inutiles et minimiser la multicolinéarité.

#### **Pertinence pour l'Analyse :**

Certaines variables peuvent être considérées comme non pertinentes pour les objectifs de l'analyse. Par exemple, des variables géographiques très détaillées comme adr (adresse), pr (point de repère), ou voie (nom de la rue) peuvent être supprimées si elles n'apportent pas de valeur ajoutée significative à l'analyse des facteurs influençant la gravité des accidents.

En appliquant cette stratégie de suppression de variables, nous nous assurons que seules les données les plus pertinentes et les plus fiables sont utilisées pour la modélisation, augmentant ainsi la qualité et l'efficacité des résultats finaux.

Voici un résumé des principales raisons pour lesquelles certaines variables ont été supprimées :

1. Dispersion trop importante ou valeurs douteuses :

- Variables géographiques détaillées : 'adr', 'com', 'dep', 'voie', 'v1', 'v2'
- Variables de localisation précise : 'lat', 'long', 'pr', 'pr1'
- Mesures de route imprécises : 'larrout', 'lartpc'

2. Redondance après transformation :

- Variables temporelles : 'an', 'jour' (utilisées pour calculer d'autres variables)
- Variables démographiques : 'an\_nais' (utilisée pour calculer l'âge)

3. Variables d'index ou d'identification :

- 'Num\_Acc', 'id\_usager', 'id\_vehicule', 'num\_veh'

4. Variables avec trop de valeurs nulles :

- 'occutc' (taux d'occupation du véhicule)

5. Variables déjà dichotomisées :

- toutes les variables qui ont été dichotomisées dans la section précédente.

#### Code Python

```

1 ######
2 # Ajout à la liste des variables trop dispersées ou douteuses pour suppression
3 #####
4 var_ecartees.append(("adr", "Dispersion trop importante"))
5 var_ecartees.append(("com", "Dispersion trop importante"))
6 var_ecartees.append(("dep", "Dispersion trop importante"))
7 var_ecartees.append(("larrout", "Valeurs douteuses"))
8 var_ecartees.append(("lartpc", "Valeurs douteuses"))
9 var_ecartees.append(("lat", "Valeurs douteuses"))
10 var_ecartees.append(("long", "Valeurs douteuses"))
11 var_ecartees.append(("occutc", "trop de nuls"))
12 var_ecartees.append(("pr", "Dispersion"))
13 var_ecartees.append(("pr1", "Dispersion"))
14 var_ecartees.append(("v1", "Dispersion"))
15 var_ecartees.append(("v2", "Dispersion"))
16 var_ecartees.append(("voie", "Dispersion"))
17 #####
18 # Ajout à la liste des variables remplacées (calculées) pour suppression
19 #####
20 var_ecartees.append(("an", "Utilisée pour déterminer les jours fériés et l'âge puis

```

```

    supprimée"))
22 var_ecartees.append(("an_nais", "Utilisée avec 'an' pour calculer l'âge puis supprimée"))
23 var_ecartees.append(("jour",      "Utilisée pour déterminer si le jour est férié puis
                           supprimée"))
24
25 ##### Ajout à la liste des variables dichotomisées pour suppression #####
26 # Ajout à la liste des variables dichotomisées pour suppression
27 ##### Les noms des variables ont déjà été ajoutées à var_ecartees #####
28 # par la fonction dichotomisation.
29
30
31 ##### Ajout à la liste des variables d'index pour suppression #####
32 # Ajout à la liste des variables d'index pour suppression
33 ##### Variables à supprimer #####
34 var_ecartees.append(("Num_Acc",      "Variable d'index"))
35 var_ecartees.append(("id_usager",    "Variable d'index"))
36 var_ecartees.append(("id_vehicule",  "Variable d'index"))
37 var_ecartees.append(("num_veh",      "Variable d'index"))
38
39 variables_a_supprimer = [ve[0] for ve in var_ecartees]
40
41 # Lors de la mise au point de ce notebook, la liste des variables à écarter
42 # est complétée avec des valeurs déjà présentes, drop est alors perturbée
43 # et ne supprime pas les variables. Il faut supprimer les doublons.
44 # C'est fait en convertissant la liste en ensemble (set)
45 # puis en la reconvertissant en liste.
46 variables_a_supprimer = set(variables_a_supprimer)
47 variables_a_supprimer = list(variables_a_supprimer)
48
49 print ("Variables à supprimer : ", variables_a_supprimer)
50 df = df.drop(columns = variables_a_supprimer, axis = 1)
51
52 print(f"Il reste les {df.shape[1]} colonnes suivantes :{", end=" "})
53 print("[ " + ", ".join(f"'{col}'" for col in df.columns) + "]")
54 print ("Valeurs nulles :\n", df.isnull().sum())

```

### 3.2.8 Suppression de doublons

La suppression de doublons est une étape essentielle pour s'assurer que chaque observation dans le jeu de données est unique et représentative. Les doublons peuvent se produire en raison de diverses erreurs de collecte ou d'intégration de données, telles que des entrées multiples pour le même accident ou des erreurs lors de la fusion des bases de données. Conserver ces doublons dans le jeu de données pourrait fausser les résultats analytiques et les prédictions des modèles en donnant un poids excessif à certaines observations.

La suppression des doublons est effectuée en identifiant les enregistrements qui ont les mêmes valeurs dans toutes les colonnes pertinentes. Après cette étape, seules les observations uniques et significatives sont conservées pour garantir la qualité et l'intégrité des données.

Code Python

```

1 # Nombre d'observations avant la suppression des doublons
2 avant_supp = df.shape[0]
3
4 # Suppression des doublons
5 df = df.drop_duplicates()
6
7 # Nombre d'observations après la suppression des doublons
8 apres_supp = df.shape[0]
9
10 # Affichage des résultats
11 print ("Suppression d'observations :")

```

```

12 print (f" avant suppression {avant_supp:7d}")
13 print (f" nous supprimons   {avant_supp - apres_supp:7d}")
14 print (f" après suppression {apres_supp:7d}")

```

### 3.2.9 Équilibrage de la dimension

L'équilibrage de la dimension est essentiel lorsque la variable cible est déséquilibrée, c'est-à-dire qu'une modalité est significativement plus représentée qu'une autre. Notre objectif étant de prédire les conditions des accidents ayant entraîné une hospitalisation de plus de 24h ou la mort, le déséquilibre entre les deux classes de la variable `grav_grave` indiquant si un accident est "grave" (1) ou "non grave" (0) doit être corrigé pour éviter que le modèle d'apprentissage soit biaisé en faveur de la classe majoritaire (les accidents "non graves"), un équilibrage de ces modalités est nécessaire.

Nous avons sous-échantillonné la classe dominante pour rééquilibrer les données avec `RandomUnderSampler` en lui précisant `random_state=8421` pour assurer la reproductibilité du processus. Le sous-échantillonnage consiste à réduire le nombre d'exemples de la classe majoritaire (ici les conséquences "non graves") afin de correspondre au nombre d'exemples de la classe minoritaire (conséquences "graves"). Cette technique permet de créer un jeu de données équilibré où les deux classes sont représentées par le même nombre d'observations.

#### Résultats de l'équilibrage :

Les nombres de modalités de notre variable cible et les nombres d'observations avant et après réduction par échantillonnage sont reportées dans le tableau suivant :

	"Graves"	"Non graves"	Total
Avant	88 821	401 827	490 648
Après	88 821	88 821	177 642

Ces valeurs issues d'un affichage dans un notebook nous assurent que nos modalités sont également réparties.

#### Code Python

```

1 ######
2 # Équilibrage des modalités de la variable cible
3 #####
4
5 df2 = df # Copie pour mise au point et reprise rapide possible du jeu de données
6
7 # Initialisation des variables pour stocker des informations
8 nb_obs = []
9 dfgs = {}
10 total = 0
11
12 # Affichage de la répartition initiale des accidents graves
13 print ("Répartition graves et autres")
14 print (df2.grav_grave.value_counts())
15
16 # Affichage du nombre total d'accidents graves
17 nb_graves = df2.grav_grave.sum()
18 print (nb_graves)
19
20 # Séparation du DataFrame en deux catégories : accidents graves et non graves
21 df_graves = df2[df2.grav_grave == 1]
22 df_autres = df2[df2.grav_grave == 0]
23
24 # Échantillon des accidents non graves pour équilibrer avec les graves
25 df_autres = df_autres.sample(nb_graves, random_state = 8421)
26
27 # Combinaison des accidents graves et de l'échantillon d'accidents non graves
28 df2 = pd.concat([df_graves, df_autres], ignore_index = True)
29
30 # Affichage de la nouvelle répartition après équilibrage
31 print ("Répartition graves et autres")
32 print (df2.grav_grave.value_counts())

```

```

33
34 # Enregistrement des actions effectuées
35 log_action(f"Équilibrage 2 parts égales dans df2")
36
37 # Remplacement du DataFrame original par la version équilibrée
38 df = df2

```

Une fois cet ensemble d'étapes effectué, nous réalisons une analyse préliminaire pour explorer le contenu de chaque colonne, notamment pour des variables binaires ou catégorielles converties en indicateurs numériques.

#### Code Python

```

1 # Nombre total d'observations dans le DataFrame
2 nb_obs = df.shape[0]
3
4 for col in df.columns: # sur chaque colonne du DataFrame
5     # Affichage du nom de la colonne, la somme et le pourcentage formaté
6     print(f"[{col:15s} {df[col].sum():6d} {100.*df[col].sum()/nb_obs:10.7f}%]")

```

Synthèse des variables clés et des statistiques principales obtenues :

Variable	Description	Pourcentage
grav_grave	Accident grave	50.000%
sexe_m	Conducteur masculin	69.271%
sexe_f	Conducteur féminin	29.985%
age_adulte	Conducteur adulte	55.641%
age_jeune	Jeune conducteur	24.473%
secu_ceinture	Port de la ceinture	49.354%
secu_casque	Port du casque	24.793%
surf_norm	Surface normale	81.101%
surf_mouil	Surface mouillée	17.164%
vma_50	Vitesse max autorisée 50 km/h	46.801%
vma_80	Vitesse max autorisée 80 km/h	20.589%
atm_1	Condition atmosphérique normale	79.979%
lum_1	Plein jour	65.967%
int_1	Hors intersection	67.038%
catu_1	Autoroute	72.507%
obsm_2	Absence d'obstacle mobile	58.188%
senc_1	Sens de circulation unique	44.430%
trajet_5	Trajet loisirs	41.870%

Interprétation :

— **Gravité des accidents :**

La variable cible 'grav\_grave' est parfaitement équilibrée à 50% grâce à l'opération d'équilibrage.

— **Caractéristiques des conducteurs :**

Les hommes sont surreprésentés (69.27%) par rapport aux femmes (29.99%). Les adultes constituent la majorité des conducteurs (55.64%), suivis par les jeunes (24.47%). Le port de la ceinture est observé dans environ la moitié des cas (49.35%). Le port du casque concerne environ un quart des cas (24.79%), probablement lié aux accidents de deux-roues.

— **Conditions de l'accident :**

La majorité des accidents se produit sur une surface normale (81.10%), avec une part non négligeable sur surface mouillée (17.16%). Les conditions atmosphériques sont généralement normales (79.98%). La plupart des accidents ont lieu en plein jour (65.97%).

— **Localisation des accidents :**

Les accidents hors intersection sont majoritaires (67.04%). Les autoroutes représentent une part importante des lieux d'accidents (72.51%).

— **Limites de vitesse :**

Les zones à 50 km/h (46.80%) et 80 km/h (20.59%) sont les plus représentées, suggérant une prédominance des accidents en ville et sur routes secondaires.

— **Autres observations notables :**

L'absence d'obstacle mobile est notée dans 58.19% des cas. Les trajets de loisirs sont les plus représentés (41.87%).

### 3.2.10 Synthèse des actions de préprocessing

La liste des actions réalisées au cours du traitement des données permet de garder une trace claire et structurée de toutes les manipulations et transformations effectuées sur le jeu de données. Dans cette section, nous effectuons un récapitulatif des étapes importantes du pipeline de prétraitement, qui peut inclure le nettoyage des données, la création de variables, l'équilibrage des classes, la suppression des doublons, et bien d'autres actions.

Code Python

```
1 # Listing de toutes les actions réalisées au cours du traitement
2 print ("Actions réalisées :")
3 for a in actions:    # sur chaque action dans la liste 'actions'
4     print (a)          # affichage de chaque action individuellement
```

Nous obtenons le résultat suivant :

Actions réalisées :

- Jointure usagers <— caractéristiques
- Jointure (usagers et caractéristiques) <— lieux
- Jointure (usagers, caractéristiques et lieux) <— véhicules
- Suppression de 0 observations dont la gravité est inconnue (codée -1)
- Création de la variable jsem : jour de la semaine
- Création de la variable ferie : jour férié, dimanches et autres fêtes
- Création de la variable age : différence entre l'année de l'accident et l'année de naissance
- Dichotomisation des champs secu1, secu2 et secu3
- Dichotomisation de l'âge
- Dichotomisation de l'heure
- Dichotomisation du sexe
- Dichotomisation de la gravité
- Dichotomisation du nombre de voies avec regroupement 1 à 4 puis 5 et plus
- Dichotomisation du l'état de la surface : sèche, mouillée, glissante (3 à 9)
- Dichotomisation de la vitesse maximale autorisée avec regroupement
- Dichotomisation en ou hors agglomération (agg), 1 agglomération, 0 hors agglomoration
- Dichotomisation de l'action du piéton (actp), modalité -1 0 et B exclues
- Dichotomisation des cond. atmosphériques (atm), modalité -1 et 9 exclues
- Dichotomisation de la catégorie de route (catr)), modalité -1 et 9 exclues
- Dichotomisation de la catégorie d'usager (catu), modalité -1 exclue
- Dichotomisation de la catégorie de véhicule (catv), modalité -1 et 0 exclues
- Dichotomisation du point de choc initial (choc), modalité -1 et 0 exclues
- Dichotomisation du régime de circulation (circ), modalité -1 exclue
- Dichotomisation du type de collision (col), modalité -1 exclue
- Dichotomisation de (etatp), modalité -1 exclue
- Dichotomisation de Aménagement - infrastructure (infra), modalités -1 et 0 exclues
- Dichotomisation du type d'intersection (int), modalité -1 exclues
- Dichotomisation du jour de la semaine (jsem), toutes modalité
- Dichotomisation de la localisation du piéton (locp), modalités -1, 0(non renseigné) exclues
- Dichotomisation des conditions lumineuses (lum), modalité -1 exclue

- Dichotomisation de la manœuvre (many), modalité -1 et 0 exclues
- Dichotomisation du mois (mois), toutes modalités
- Dichotomisation de la motorisation (motor), modalité -1 et 0 exclues
- Dichotomisation de l'obstacle fixe heurté (obs), modalité -1 et 0 exclues
- Dichotomisation de l'obstacle mobile heurté (obsm), modalité -1 et 0 exclues
- Dichotomisation de la place dans le véhicule (place), modalité -1 exclue
- Dichotomisation du tracé en plan (plan), modalité -1 exclue
- Dichotomisation de la déclivité (prof), modalité -1 exclue
- Dichotomisation du sens de circulation (senc), modalité -1, 0 et 3 exclues
- Dichotomisation de la situation de l'accident (situ), modalité -1 exclue
- Dichotomisation du motif du trajet (trajet), modalité -1 exclue
- Dichotomisation de la présence d'une voie réservée (vosp), modalités -1 exclue
- Équilibrage 2 parts égales dans df2

### 3.2.11 Sauvegarde des données

Une fois que toutes les étapes de nettoyage, transformation, équilibrage, et ingénierie des variables ont été réalisées, il est nécessaire de sauvegarder le jeu de données final ainsi que la documentation associée. Ces enregistrements permettent de garantir que les données et les métadonnées sont prêtes et disponibles pour les analyses ou modélisations futures.

Code Python

```

1 ######
2 # Enregistrements :
3 #   - Le jeu de données préparé pour la modélisation ;
4 #   - La description des variables mise à jour.
5 #####
6
7 df.to_csv(rep_dst + '/' + "data.csv", sep = '\t', index=False, encoding='utf-8')
8
9 with open("./desc_vars.json", 'w', encoding='utf-8') as fichier:
10    json.dump(desc_vars, fichier, ensure_ascii=True, indent=True)

```

## 3.3 Modélisation 2

- Importation des librairies

Toutes les bibliothèques nécessaires à la manipulation des données sont préalablement chargées.

Code Python

```

1 import numpy as np
2 import pandas as pd
3 import time
4 from datetime import datetime
5 import joblib
6 import os
7 from tabulate import tabulate
8 from sklearn.decomposition import PCA
9 from sklearn.model_selection import GridSearchCV
10 from sklearn.svm import SVC
11 from sklearn.svm import LinearSVC
12 from lightgbm import LGBMClassifier
13 from sklearn.preprocessing import StandardScaler
14 from sklearn.model_selection import train_test_split
15 from sklearn.neighbors import KNeighborsClassifier
16 from sklearn.dummy import DummyClassifier
17 from sklearn.tree import DecisionTreeClassifier

```

```

18 from sklearn.linear_model           import LogisticRegression
19 from sklearn.ensemble              import HistGradientBoostingClassifier
20 from sklearn.ensemble              import GradientBoostingClassifier
21 from sklearn.ensemble              import BaggingClassifier
22 from sklearn.ensemble              import ExtraTreesClassifier
23 from sklearn.ensemble              import ExtraTreesRegressor
24 from sklearn.model_selection       import cross_validate
25 from sklearn.model_selection       import cross_val_score
26 from sklearn.metrics               import f1_score
27 from sklearn.metrics               import r2_score
28 from sklearn.metrics               import confusion_matrix
29 from sklearn.metrics               import classification_report
30 from sklearn.metrics               import recall_score
31 from sklearn.metrics               import fbeta_score
32 from sklearn.metrics               import precision_score
33 from sklearn.metrics               import roc_auc_score
34 from sklearn.feature_selection    import f_regression
35 from sklearn.ensemble              import RandomForestClassifier
36 import json
37 from matplotlib import pyplot as plt
38 import seaborn as sns
39
40 rep_dst = "data/processed/" # Fichiers utilisés par la modélisation

```

- Fonction d'affichage des messages

La fonction qui suit affiche les messages à l'écran et les ajoute à une chaîne de caractères nommée 'logres'. Elle permet d'enregistrer les résultats d'entraînement dans un fichier.

#### Code Python

```

1 logres = ""
2 def printlog(msg=""):
3     global logres
4     logres += f"{msg}\n"
5     print (msg)
6     return

```

- Fonction de formatage de la durée

La fonction qui suit convertit une durée en une chaîne de caractères contenant des heures, minutes et secondes.

#### Code Python

```

1 def format_duree(seconds_f):
2     # Conversion en heures, minutes et secondes
3     seconds = int(seconds_f)
4     hours = seconds // 3600
5     minutes = (seconds % 3600) // 60
6     secs = seconds % 60
7
8     # Formatage conditionnel en fonction de la durée
9     if hours > 0:
10         return f"{hours} h {minutes}min"
11     elif minutes > 0:
12         return f"{minutes} min {secs} s"
13     else:
14         return f"{secs} s"

```

- Chargement et préparation des données

Le code suivant charge le jeu de données et le sépare en variables explicatives (X) et variable cible (y) nommée 'grav\_grave'.

#### Code Python

```
1 # Lecture du jeu de données
2 df = pd.read_csv(rep_dst + '/' + "data.csv", sep = '\t', index_col = None)
3
4 # Séparation des variables explicatives et de la variable cible
5 X = df.drop(['grav_grave'], axis = 1)
6 feature_names = X.columns
7
8 y = df.grav_grave
```

- Réduction de dimension et standardisation des données

Le jeu de données contient plus de 250 variables explicatives. Ce nombre étant très important, nous décidons d'appliquer une PCA. Après avoir appliqué la PCA, nous normalisons le jeu de données de dimension réduite. Deux graphiques nous aident à choisir le nombre de composantes en affichant la variance expliquée par composante.

#### Code Python

```
1 # Réduction de dimension
2 pca = PCA(n_components = 100)
3 pca.fit(X)
4
5 plt.figure()
6 plt.xlim(0,X.shape[1])
7 plt.plot(pca.explained_variance_ratio_)
8 plt.title('Variance expliquée par composante');
9
10 plt.figure()
11 plt.xlim(0,X.shape[1])
12 plt.xlabel('Nombre de composantes')
13 plt.ylabel('Part de variance expliquée')
14 plt.axhline(y = 0.9, color ='r', linestyle = '--')
15 plt.plot(pca.explained_variance_ratio_.cumsum())
16 plt.title('Cumul de la variance expliquée');
17
18 X = pca.transform(X)
19
20 # standardisation
21 scaler = StandardScaler()
22 X = scaler.fit_transform(X)
```

Nous obtenons les résultats suivants :

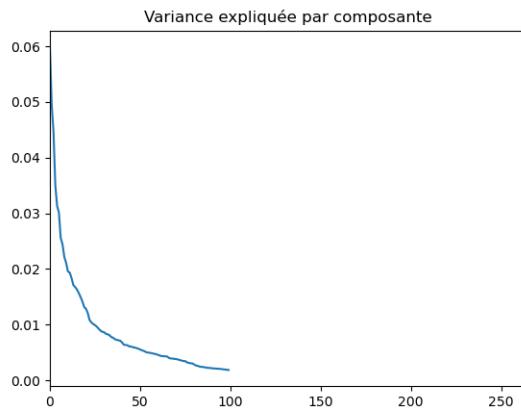


FIGURE 1 – Variance Expliquée

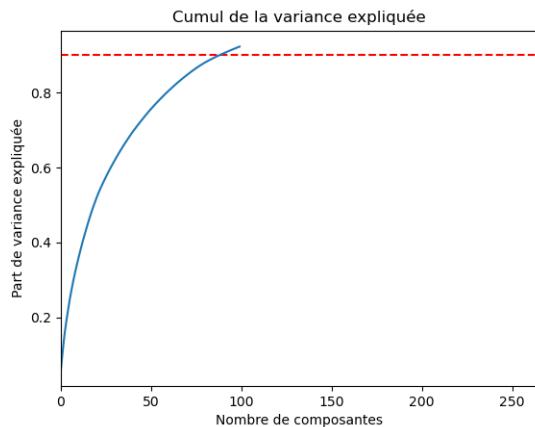


FIGURE 2 – Cumul de Variance Expliquée

- Séparation du jeu de données

Les données sont divisées en ensemble d'entraînement et de test, et le random\_state est fixé à 1234 pour assurer la reproductibilité de cette division.

#### Code Python

```
1 # Séparation du jeu de données d'entraînement et de test
2 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = .20, random_state =
1234)
```

- Définition des modèles

Le dictionnaire suivant fixe la liste des modèles à implémenter sur la base de quelques éléments mentionnés ci-après :

- "prmsgscv" : paramètres à optimiser via une grille de recherche (GridSearchCV).
- "prmfixes" : paramètres fixes du modèle.
- "perf" : destiné à stocker les performances du modèle.
- "nom" : nom court du modèle.
- "libelle" : description du modèle.
- "classe" : chemin d'importation de la classe du modèle dans scikit-learn.
- "instance" : instance du modèle avec des paramètres par défaut ou spécifiques.

#### Code Python

```
1 modeles = {
2
3     "SVC" : {
4         "prmsgscv" : {"gamma" : ["scale", "auto"]}
5                 },
6         "prmfixes" : {},
7         "pred" : None,
8         "perf" : [],
9         "nom" : "SVC",
10        "libelle" : "Classification à support de vecteurs",
11        "classe" : "sklearn.svm.SVC",
12        "instance" : SVC(),
13        "grid" : None
14    },
15
16     "LR" : {
17         "prmsgscv" : {'solver' : ['liblinear', 'lbfgs']},
18         "prmfixes" : {}}
```

```

18           'C'      : [0.003, 0.005, 0.01, 0.02, 0.04]},  

19   "prmfixes" : {},  

20   "perf"     : {},  

21   "nom"      : "LogisticRegression",  

22   "libelle"  : "Régression logistique",  

23   "classe"   : "sklearn.xxx",  

24   "instance" : LogisticRegression(max_iter = 10000),  

25   "grid"     : None  

26 },  

27  

28 "HGB" : {  

29   "prmsgscv" : {"max_leaf_nodes" : [None, 15, 31, 63 ],  

30                 "l2_regularization" : [0.001, 0.01, 0.1, 1, 10],  

31                 },  

32   "prmfixes" : {},  

33   "perf"     : {},  

34   "nom"      : "HistGradientBoostingClassifier",  

35   "libelle"  : "Hist Gradient Boosting Classifier",  

36   "classe"   : "sklearn.neighbors.HistGradientBoostingClassifier",  

37   "instance" : HistGradientBoostingClassifier(random_state = 421),  

38   "grid"     : None  

39 },  

40  

41 "LGBM" : {  

42   "prmsgscv" : {"boosting_type" : ["gbdt", "dart", "goss"],  

43                 },  

44   "prmfixes" : {},  

45   "perf"     : {},  

46   "nom"      : "LGBMClassifier",  

47   "libelle"  : "LGBMClassifier",  

48   "classe"   : "sklearn.",  

49   "instance" : LGBMClassifier(verbosity=-1, random_state = 421),  

50   "grid"     : None  

51 },  

52  

53 "DT" : {  

54   "prmsgscv" : {"criterion" : ["gini", "entropy", "log_loss"],  

55                 "splitter" : ["best", "random"]},  

56  

57   "prmfixes" : {},  

58   "perf"     : {},  

59   "nom"      : "DecisionTreeClassifier",  

60   "libelle"  : "Arbre de décision",  

61   "classe"   : "sklearn.tree.DecisionTreeClassifier",  

62   "instance" : DecisionTreeClassifier(random_state = 123),  

63   "grid"     : None  

64 },  

65  

66 "ET" : {  

67   "prmsgscv" : {},  

68   "prmfixes" : {},  

69   "perf"     : {},  

70   "nom"      : "ExtraTreesClassifier",  

71   "libelle"  : "Extra Trees Classifier",  

72   "classe"   : "sklearn.ensemble.ExtraTreesClassifier",  

73   "instance" : ExtraTreesClassifier(),  

74   "grid"     : None  

75 },  

76  

77 },  

78

```

```

79 "BAG" : {
80     "prmsgscv" : {},
81     "prmfixes" : {},
82     "perf" : {},
83     "nom" : "BaggingClassifier",
84     "libelle" : "Bagging Classifier",
85     "classe" : "sklearn.neighbors.BaggingClassifier",
86     "instance" : BaggingClassifier(),
87     "grid" : None
88 },
89
90 "KNN" : {
91     "prmsgscv" : {"metric" : ['minkowski', 'manhattan', 'chebyshev']},
92     "prmfixes" : {},
93     "perf" : {},
94     "nom" : "KNeighborsClassifier",
95     "libelle" : "Plus proches voisins",
96     "classe" : "sklearn.neighbors.KNeighborsClassifier",
97     "instance" : KNeighborsClassifier(),
98     "grid" : None
99 },
100
101 "RF" : {
102     "prmsgscv" : {},
103     "prmfixes" : {},
104     "perf" : {},
105     "nom" : "RandomForestClassifier",
106     "libelle" : "Forêt aléatoire",
107     "classe" : "sklearn.neighbors.RandomForestClassifier",
108     "instance" : RandomForestClassifier(random_state = 421),
109     "grid" : None
110 },
111
112 "GB" : {
113     "prmsgscv" : {"loss" : ["log_loss", "exponential"],
114                   "criterion" : ["friedman_mse", "squared_error"],
115                   },
116     "prmfixes" : {},
117     "perf" : {},
118     "nom" : "GradientBoostingClassifier",
119     "libelle" : "Gradient Boosting Classifier",
120     "classe" : "sklearn.neighbors.GradientBoostingClassifier",
121     "instance" : GradientBoostingClassifier(random_state = 421),
122     "grid" : None
123 },
124
125
126 # Classificateur bidon servant de comparaison aux autres modèles.
127 "BIDON" : {
128     "prmsgscv" : {},
129     "prmfixes" : {},
130     "perf" : {},
131     "nom" : "DummyClassifier",
132     "libelle" : "Dummy Classifier",
133     "classe" : "sklearn.dummy.DummyClassifier",
134     "instance" : DummyClassifier(random_state = 421, strategy = "uniform"),
135     "grid" : None
136 }
137
138 }

```

Voici le résultat partiellement obtenu :

Tailles	Total	Graves (True)		Non graves (False)	
		Nombre	Prop	Nombre	Prop
Jeu d'entraînement	142113	71101	50.03%	71012	49.97%
Jeu de test	35529	17720	49.87%	17809	50.13%
Total	177642	88821	50.00%	88821	50.00%

Nombre de variables explicatives (après PCA) : 100

- - - Classification à support de vecteurs [1/11] - - -

Paramètres essayés :

gamma : ['scale', 'auto']

SVC()

Entraînement avec GridSearchCV : Recherche des meilleurs paramètres

Paramètres retenus :

gamma : scale

Prédiction :

Durée	:	1 h 43min
Score sur le jeu d'entraînement	:	0.8491%
Score sur le jeu de test	:	0.7828
Scores F1	:	0.7717, 0.7928

	precision	recall	f1-score	support
False	0.82	0.73	0.77	17809
True	0.76	0.83	0.79	17720
accuracy			0.78	35529
macro avg	0.79	0.78	0.78	35529
weighted avg	0.79	0.78	0.78	35529

Pas de feature\_importances

Matrice de confusion :

```
[[ 13047  4762 ]
 [ 2956 14764 ]]
```

```
[[ 0.36722114  0.13403135 ]
 [ 0.08319964  0.41554786 ]]
```

Enregistrement du modèle entraîné : SVC.mdl, 65930443octets, 62Mo

(...)

- Initialisation de comptes-rendus d'analyse

Le code suivant crée un DataFrame tbl\_perf pour stocker les performances des modèles, avec des colonnes pour différentes métriques (score, durée, précision, rappel, etc.).

Egalement, il calcule et stocke des statistiques sur les ensembles de données : nombre total d'observations, nombre d'observations pour l'entraînement et le test, nombre d'observations "graves" et "non graves" pour chaque ensemble, etc.

Il affiche un tableau récapitulatif des statistiques calculées, montrant la répartition des classes dans les ensembles d'entraînement et de test.

Enfin, il initialise un compteur de temps (ttot1) et un index de modèle (idx\_modele).

## Code Python

```

1 ##### Initialisations : variables, comptes-rendus, tables...
2 # Initialisations : variables, comptes-rendus, tables...
3 #####
4
5 # Tableau des performances
6 tbl_perf = pd.DataFrame({ "Modele":[] , "libelle":[] ,
7                           "duree_GSCV": [] , "duree_GSCV_str": [] ,
8                           "score_ent" : [] , "score_test":[] ,
9                           "score_f1_0":[] , "score_f1_1":[] ,
10                          "score_auc":[] ,
11                          "score_precision":[] ,
12                          "score_recall_0":[] , "score_recall_1":[] ,
13                          "score_fbeta_b2":[] ,
14                          "tn":[] , "fp":[] , "fn":[] , "tp":[] ,
15                          "rate_tn":[] , "rate_fp":[] , "rate_fn":[] , "rate_tp":[] ,
16                          "taille_mdl":[]
17 })
18
19 # logres est une chaîne de caractères contenant les messages destinés
20 # à l'affichage sur le notebook et l'enregistrement dans un fichier texte.
21
22 logres = "" # Effacement des messages, en cas de ré-exécution de la cellule
23 printlog(datetime.now())
24
25 # Observations statistiques
26 nb_tot = int(X_test.shape[0] + X_train.shape[0]) # Nbre total d'observations
27 nb_ent = int(X_train.shape[0]) # Nbre d'observations pour les
28                                entrainements
29 nb_test = int(X_test.shape[0]) # Nbre d'observations pour les tests
30
31 nb_tot_true = int(y_test.sum() + y_train.sum()) # Nbre total d'observation 'graves'
32 nb_ent_true = int(y_train.sum()) # Nbre d'observation 'graves' pour les
33                                entrainements
34 nb_test_true = int(y_test.sum()) # Nbre d'observation 'graves' pour les
35                                tests
36
37 nb_tot_false = nb_tot - nb_tot_true # Nbre total d'observation 'non graves'
38 nb_ent_false = nb_ent - nb_ent_true # Nbre d'observation 'non graves' pour
39                                les entrainements
40 nb_test_false = nb_test - nb_test_true # Nbre d'observation 'non graves' pour
41                                les tests
42
43 # Tableau d'affichage des statistiques
44 printlog (f" Tailles          | Total | graves (True) | Non graves (False)")
45 printlog (f"                  |       | nombre | prop    | nombre | prop")
46 printlog (f"Jeu d'entraînement | {nb_ent:6d} | {nb_ent_true:6d} |
47             {100.*nb_ent_true/nb_ent:6.2f}\% | {nb_ent_false:6d} |
48             {100.*nb_ent_false/nb_ent:6.2f}\%")
49 printlog (f"Jeu de test      | {nb_test:6d} | {nb_test_true:6d} |
50             {100.*nb_test_true/nb_test:6.2f}\% | {nb_test_false:6d} |
51             {100.*nb_test_false/nb_test:6.2f}\%")
52 printlog (f"Total           | {nb_tot:6d} | {nb_tot_true:6d} |
53             {100.*nb_tot_true/nb_tot:6.2f}\% | {nb_tot_false:6d} |
54             {100.*nb_tot_false/nb_tot:6.2f}\%")
55 printlog ()
56 printlog (f"Nombre de variables explicatives (après PCA) : {X_train.shape[1]:6d}")
57
58 # Suivi chronométré et numéroté des modèles
59 ttot1 = time.time()
60 idx_modele=1 # Numéro du modèle en cours, pour l'affichage uniquement

```

Voici un extrait des résultats obtenus :

	Libellé	Entraînement	Test	f1 0	f1 1	AUC
0	Classification à support de vecteurs	0.849148	0.782769	0.771738	0.792783	0.782895
1	Régression logistique	0.755582	0.752568	0.749108	0.755934	0.752608
2	Hist Gradient Boosting Classifier	0.974534	0.774128	0.766314	0.781436	0.774217

	Libellé	Précision	Recall 0	Recall 1	f béta=2
0	Classification à support de vecteurs	0.785702	0.732607	0.833183	0.816539
1	Régression logistique	0.752835	0.736931	0.768284	0.763296
2	Hist Gradient Boosting Classifier	0.775541	0.738840	0.809594	0.798091

	Libellé	Vrais nég.	Faux pos.	Faux nég.	Vrais pos.
0	Classification à support de vecteurs	13047	4762	2956	14764
1	Régression logistique	13124	4685	4106	13614
2	Hist Gradient Boosting Classifier	13158	4651	3374	14346

	Libellé	% Vrais nég.	% Faux pos.	% Faux nég.	% Vrais pos.
0	Classification à support de vecteurs	0.367221	0.134031	0.0831996	0.415548
1	Régression logistique	0.369388	0.131864	0.115568	0.38318
2	Hist Gradient Boosting Classifier	0.370345	0.130907	0.0949647	0.403783

	Libellé	Durée entraînement	Taille modèle
0	Classification à support de vecteurs	6229.81	6.59304e+07
1	Régression logistique	0.501175	1647
2	Hist Gradient Boosting Classifier	90.3109	2.41198e+07

- Optimisation des modèles et évaluation des performances

Le code suivant implémente une boucle d'entraînement et d'évaluation pour les modèles de classification figurant dans le dictionnaire préalablement défini.

Pour chaque modèle, il utilise GridSearchCV pour optimiser les hyperparamètres, puis entraîne le modèle avec la meilleure configuration.

Il effectue ensuite des prédictions sur un ensemble de test et calcule diverses métriques de performance, telles que le score F1, l'AUC, la précision et le rappel.

Le code affiche également des informations détaillées comme la matrice de confusion et l'importance des variables (si disponible).

Les performances de chaque modèle sont stockées dans le DataFrame tbl\_perf pour une comparaison facile.

Enfin, le code enregistre chaque modèle entraîné sur le disque et génère un rapport complet dans un fichier texte.

#### Code Python

```

1 ######
2 # Boucle pour chaque modèle
3 #####
4
5 for k, m in modeles.items():
6     perf = {}
7     printlog()
8     printlog(f"----- {m['libelle'][:20s]} [{idx_modele:2d}/{len(modeles):2d}] -----")
9     printlog("")
10    idx_modele += 1
11    model = m["instance"]
12    params = m["prmsgscv"]
13    printlog(f"Paramètres essayés : ")

```

```

14     for prm, val in params.items():
15         printlog (f"      {prm:15s} : {val}")
16     printlog ()
17     printlog (model)
18
19 ##### Essais avec GridSearchCV et rech., le cas échéant, meilleurs paramètres #####
20 # Essais avec GridSearchCV et rech., le cas échéant, meilleurs paramètres
21 #####
22
23     printlog ("Entraînement avec GridSearchCV : Recherche des meilleurs paramètres")
24
25     t1 = time.time()
26
27     grille_clf = GridSearchCV(model, param_grid = params, cv = 5, n_jobs = -1)
28     grille = grille_clf.fit (X_train, y_train)
29
30     t2 = time.time()
31
32     printlog ()
33     printlog ("Paramètres retenus :")
34     for it in grille.best_params_.items():
35         printlog (f"  {it[0]:20s} : {it[1]}")
36     printlog ()
37
38     # Récupération du meilleur modèle
39     model = grille_clf.best_estimator_
40
41 ##### Prédition avec le modèle sélectionné #####
42 # Prédition avec le modèle sélectionné
43 #####
44
45     printlog ("Prédition :")
46     y_pred = model.predict (X_test)
47     t = time.time()
48
49 ##### Calculs des scores, métriques et autres valeurs #####
50 # Calculs des scores, métriques et autres valeurs
51 #####
52
53 # Le temps d'entraînement d'un modèle avec les meilleurs paramètres n'est
54 # pas connu. Le temps retenu est le temps d'entraînement d'un modèle
55 # ou de plusieurs modèles avec paramètres différents.
56 # Il est affiché pour information.
57
58     duree_GSCV = grille_clf.refit_time_          # temps d'entraînement du meilleur modèle
59     duree_GSCV_str = format_duree(duree_GSCV) # le même plus lisible : h m s
60     score_ent = model.score(X_train, y_train)
61     score_test = model.score(X_test, y_test)
62     score_f1_0, score_f1_1 = f1_score(y_test, y_pred, average = None)
63     score_auc = roc_auc_score(y_test, y_pred, average = None)
64     score_precision = precision_score(y_test, y_pred, average = 'macro')
65     score_recall_0, score_recall_1 = recall_score(y_test, y_pred, average = None)
66     score_fbeta_b2 = fbeta_score(y_test, y_pred, beta = 2)
67
68     printlog (f"Durée                  : {duree_GSCV_str}")
69     printlog (f"Score sur le jeu d'entraînement : {score_ent:7.4f}\%")
70     printlog (f"Score sur le jeu de test       : {score_test:7.4f}")
71     printlog (f"Scores f1                 : {score_f1_0:7.4f} {score_f1_1:7.4f}")
72
73     printlog()
74     printlog (classification_report(y_test, y_pred))

```

```

75     printlog ()
76
77     try:
78         fimp = model.feature_importances_
79         classement_var = pd.DataFrame(data={"Variable" : feature_names, "description" :
80             "feature_full_names, "Importance" : model.feature_importances_})
81         classement_var = classement_var.sort_values (by = "Importance", ascending =
82             False)
83         printlog (f"Importance des variables :")
84         printlog (classement_var.to_string())
85
86     except AttributeError:
87         printlog ("Pas de feature_importances_")
88     except Exception:
89         printlog ("feature importances non utilisable")
90
91     printlog ()
92     printlog ("Matrice de confusion :")
93     ctab = pd.crosstab(y_test, y_pred, rownames=['Classe réelle'], colnames=['Classe
94         prédictive'])
95     cm = confusion_matrix(y_test, y_pred)
96     printlog (cm)
97     tn, fp, fn, tp = cm[0,0], cm[0,1], cm[1,0], cm[1,1]
98     printlog (confusion_matrix(y_test, y_pred, normalize = "all"))
99     printlog ()
100    nom_fic_mdl = m["nom"]+".mdl"
101    joblib.dump(model, nom_fic_mdl)
102    taille_mdl = os.path.getsize(nom_fic_mdl)
103    printlog (f"Enregistrement du modèle entraîné : {nom_fic_mdl} , {taille_mdl}octets,
104        {int(os.path.getsize(nom_fic_mdl)/(1024*1024))}Mo")
105
106 #####
107 # Stockage des performances dans le DataFrame tbl_perf
108 #####
109 rate_tn, rate_fp, rate_fn, rate_tp = tn/nb_test, fp/nb_test, fn/nb_test, tp/nb_test
110
111 perf_modele = pd.DataFrame([{"Modele":k, "libelle":m["libelle"],
112                             "duree_GCSV": duree_GCSV, "duree_GCSV_str": duree_GCSV_str,
113                             "score_ent" : score_ent, "score_test":score_test,
114                             "score_f1_0":score_f1_0, "score_f1_1":score_f1_1,
115                             "score_auc":score_auc,
116                             "score_precision":score_precision,
117                             "score_recall_0":score_recall_0,
118                             "score_recall_1":score_recall_1,
119                             "score_fbeta_b2":score_fbeta_b2,
120                             "tn" : tn, "fp" : fp, "fn" : fn, "tp": tp,
121                             "rate_tn" : rate_tn, "rate_fp" : rate_fp, "rate_fn" : rate_fn,
122                             "rate_tp": rate_tp,
123                             "taille_mdl":taille_mdl
124                         }])
125
126     # Ajouter la nouvelle ligne avec concat
127     tbl_perf = pd.concat([tbl_perf, perf_modele], ignore_index=True)
128
129 ttot2 = time.time()
130 printlog()
131 printlog()
132 printlog(str(tbl_perf))
133 printlog()
134 printlog()
135 printlog (f"Temps total {int(ttot2-ttot1)//60} min {int(ttot2-ttot1)%60} s.")

```

```

130 printlog()
131 printlog(datetime.now())
132 with open("entraînement.txt", "wt") as file:
133     file.write (logres)

```

- Affichage des performances

Le code qui suit utilise la fonction 'tabulate' pour afficher de manière formatée et lisible les performances des différents modèles de classification stockées dans le DataFrame 'tbl\_perf'. Il crée cinq tableaux distincts :

- Le premier tableau montre les scores d'entraînement, de test, les scores F1 pour les classes 0 et 1, et l'AUC.
- Le deuxième tableau présente la précision, le recall pour les classes 0 et 1, et le score F-beta avec beta=2.
- Le troisième tableau affiche les valeurs brutes de la matrice de confusion (vrais négatifs, faux positifs, faux négatifs, vrais positifs).
- Le quatrième tableau montre les mêmes informations que le précédent, mais en pourcentages.
- Le dernier tableau indique la durée d'entraînement (en secondes et en format lisible) ainsi que la taille du modèle enregistré.

#### Code Python

```

1 print (tabulate (tbl_perf[['libelle', 'score_ent', 'score_test', 'score_f1_0',
2     'score_f1_1', 'score_auc']],
3     headers = ["libellé", "Entraînement", "Test", "f1 0", "f1 1 ", "AUC"  ]))
4 print()
5 print (tabulate (tbl_perf[['libelle', 'score_precision', 'score_recall_0',
6     'score_recall_1', 'score_fbeta_b2']],
7     headers = ["libellé", "Précision", "Recall 0", "Recall 1", "f béta=2"]))
8 print()
9 print (tabulate (tbl_perf[['libelle', 'tn', 'fp', 'fn', 'tp']],
10    headers = ['libellé', 'Vrais nég.', 'Faux pos.', 'Faux nég.', 'Vrais pos.']))
11 print()
12 print()
13 print (tabulate (tbl_perf[['libelle', 'rate_tn', 'rate_fp', 'rate_fn', 'rate_tp']],
14    headers = ['libellé', '\%_Vrais nég.', '\%_Faux pos.', '\%_Faux nég.', '\%_Vrais
15    pos.']))
16 print()
17 print (tabulate (tbl_perf[['libelle', 'duree_GSCV', 'duree_GSCV_str', 'taille_mdl']],
18    headers = ["libellé", "Durée entraînement", "Id.", "Taille modèle"]))

```

- Graphique de comparaison des scores d'entraînement et de test par modèle

Le code suivant crée un graphique à barres comparant les scores d'entraînement (en bleu) et de test (en vert) pour les différents modèles de classification. Une ligne horizontale en pointillés noirs indique le meilleur score de test obtenu.

#### Code Python

```

1 fig = plt.figure(figsize = (10, 4))
2
3 bar_width = 0.4
4 x_indexes = np.arange(tbl_perf.shape[0])
5 x_indexes_z = x_indexes + bar_width # Décalage pour les barres de z
6
7 plt.bar(x_indexes, tbl_perf["score_ent"]*100., width=bar_width, color='blue',
8         label='Entraînement')
8 plt.bar(x_indexes_z, tbl_perf["score_test"]*100., width=bar_width, color='green',
9         label='Test')
9 meilleur_score = tbl_perf["score_test"].max()*100

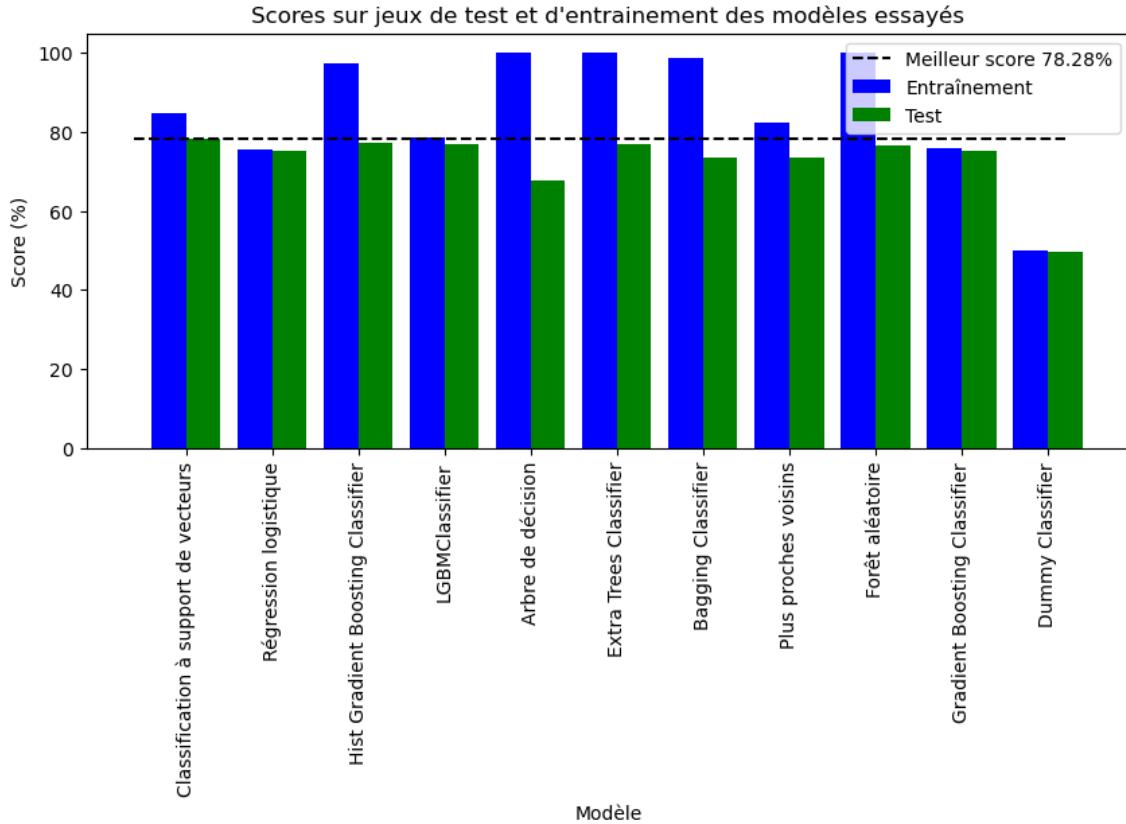
```

```

10 plt.hlines(meilleur_score, x_indexes[0] - bar_width, x_indexes[-1] + bar_width,
11             color='black', linestyles='dashed', label=f"Meilleur score
12             {meilleur_score:.2f}%)")
13 plt.title("Scores sur jeux de test et d'entraînement des modèles essayés")
14 plt.xlabel("Modèle")
15 plt.ylabel("Score (%)")
16 plt.xticks(x_indexes + bar_width / 2, tbl_perf["libelle"], rotation = 90)
17 plt.legend()
18 plt.show()

```

Nous obtenons le résultat suivant :



- Graphique d'évaluation des performances par modèle

Le code suivant crée un graphique à barres empilées comparant les performances des différents modèles de classification. Il génère quatre séries de barres côté à côté, représentant les vrais positifs (vert), faux positifs (orange), vrais négatifs (bleu) et faux négatifs (rouge) pour chaque modèle. Une ligne horizontale en pointillés noirs indique le minimum de faux négatifs parmi les modèles (excluant le "Dummy Classifier").

Code Python

```

1 fig = plt.figure(figsize = (10, 4))
2
3 bar_width = 0.2
4 x1 = np.arange(tbl_perf.shape[0]) - bar_width
5 x2 = x1 + 1 * bar_width
6 x3 = x1 + 2 * bar_width
7 x4 = x1 + 3 * bar_width
8
9 plt.bar(x1, 100.*tbl_perf["tp"]/nb_test, width=bar_width, color='green', label='vrais
10         positifs')
11 plt.bar(x2, 100.*tbl_perf["fp"]/nb_test, width=bar_width, color='orange', label='Faux
12         négatifs')
13 plt.bar(x3, 100.*tbl_perf["tn"]/nb_test, width=bar_width, color='blue', label='vrais
14         négatifs')
15 plt.bar(x4, 100.*tbl_perf["fn"]/nb_test, width=bar_width, color='red', label='Faux
16         positifs')
17 plt.hlines(meilleur_score, x1 - bar_width, x4 + bar_width, color='black', linestyles='dashed',
18             label=f"Meilleur score {meilleur_score:.2f} %")
19 plt.title("Scores sur jeux de test et d'entraînement des modèles essayés")
20 plt.xlabel("Modèle")
21 plt.ylabel("Score (%)")
22 plt.xticks(x1 + bar_width / 2, tbl_perf["libelle"], rotation = 90)
23 plt.legend()
24 plt.show()

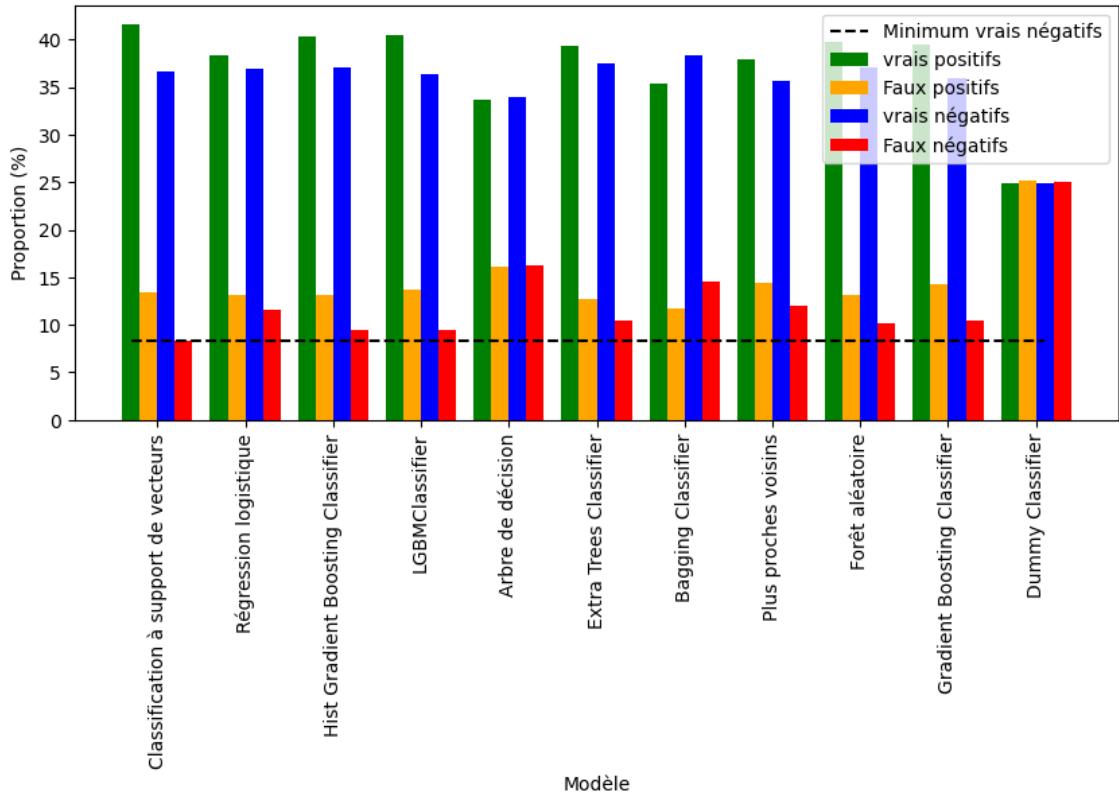
```

```

    positifs')
11 plt.bar(x3, 100.*tbl_perf["tn"]/nb_test, width=bar_width, color='blue', label='vrais
    négatifs')
12 plt.bar(x4, 100.*tbl_perf["fn"]/nb_test, width=bar_width, color='red', label='Faux
    négatifs')
13
14 min_fn = 100.*tbl_perf.loc[tbl_perf["libelle"] != "Dummy Classifier", "fn"].min()/nb_test
15 plt.hlines(min_fn, x_indexes[0] - bar_width, x_indexes[-1] + bar_width,
16             color='black', linestyles='dashed', label="Minimum vrais négatifs")
17 plt.title("")
18 plt.xlabel("Modèle")
19 plt.ylabel("Proportion (%)")
20 plt.xticks(x_indexes + bar_width / 2, tbl_perf["libelle"], rotation = 90)
21 plt.legend()
22
23 plt.show()

```

Nous obtenons le résultat suivant :



### 3.4 Modélisation 3

- Importation des librairies

Toutes les bibliothèques nécessaires à la manipulation des données sont préalablement chargées.

Code Python

```

1 import pandas as pd
2 import numpy as np
3 import seaborn as sns
4 import matplotlib.pyplot as plt
5 from sklearn.model_selection import train_test_split, GridSearchCV
6 from sklearn.preprocessing import StandardScaler

```

```

7 from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score
8 from scikeras.wrappers import KerasClassifier
9 from tensorflow.keras.models import Sequential, Model
10 from tensorflow.keras.layers import Dense, Dropout
11 from tensorflow.keras.optimizers import Adam, RMSprop
12 from sklearn.metrics import roc_curve, auc
13 import warnings

```

- Filtrage de warnings spécifiques

Le code suivant vise à filtrer et ignorer l'avertissement spécifié pour réduire le "bruit" au niveau de la sortie du programme.

Code Python

```

1 warnings.filterwarnings("ignore", message="Do not pass an `input_shape`/`input_dim`"
                         argument to a layer")

```

- Chargement et préparation des données

Le code suivant procède au chargement de l'ensemble de données, puis prépare ces données. Il sépare donc les variables explicatives (features) de la variable cible 'grav\_grave', divise les données en ensembles d'entraînement et de test, et normalise les caractéristiques à l'aide de StandardScaler.

Code Python

```

1 rep_dst = "data/processed/" # Fichier utilisé pour la modélisation
2
3 # Chargement des données
4 df = pd.read_csv(rep_dst + '/' + "data.csv", sep = '\t', index_col = None)
5 df = df.astype(int)
6
7 # Séparation des features et de la variable cible
8 X = df.drop('grav_grave', axis=1)
9 y = df['grav_grave']
10
11 # Division en ensembles d'entraînement et de test avec stratification
12 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                                       random_state=42, stratify=y)
13
14 # Normalisation des features
15 scaler = StandardScaler()
16 X_train_scaled = scaler.fit_transform(X_train)
17 X_test_scaled = scaler.transform(X_test)

```

- Préparation du modèle

Le code qui suit définit une fonction 'create\_model' qui construit un réseau de neurones séquentiel pour la classification binaire, avec des couches Dense et Dropout, en utilisant Keras. Il crée ensuite un wrapper KerasClassifier autour de ce modèle, permettant son utilisation avec des outils de scikit-learn comme GridSearchCV pour l'optimisation des hyperparamètres. Cette approche vise à permettre la combinaison des techniques traditionnelles de machine learning et avec celles du deep learning.

Code Python

```

1 # Obtenir le nombre correct de features
2 input_shape = X_train_scaled.shape[1]
3
4 # Fonction pour construire le modèle Keras
5 def create_model(optimizer='adam', init='glorot_uniform'):

```

```

6   model = Sequential([
7       Dense(64, activation='relu', input_shape=(input_shape,),
8             kernel_initializer=init),
9       Dropout(0.3),
10      Dense(32, activation='relu', kernel_initializer=init),
11      Dropout(0.3),
12      Dense(16, activation='relu', kernel_initializer=init),
13      Dropout(0.3),
14      Dense(1, activation='sigmoid')
15  ])
16  model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
17  return model
18 # Création d'un wrapper KerasClassifier pour utiliser GridSearchCV
19 model = KerasClassifier(
20     model=create_model,
21     verbose=0,
22     optimizer='adam',
23     epochs=50,
24     batch_size=32,
25     model__init='glorot_uniform'
26 )

```

- Configuration et exécution de la recherche d'hyperparamètres

Ce code utilise GridSearchCV pour tester systématiquement différentes combinaisons d'hyperparamètres (taille des lots, époques, optimiseur, initialisation) sur un modèle de réseau de neurones. Il entraîne le modèle avec chaque combinaison, utilisant une validation croisée à 3 plis, puis identifie et affiche la configuration qui donne les meilleures performances.

#### Code Python

```

1 # Définition des hyperparamètres à tester avec GridSearchCV
2 param_grid = {
3     'batch_size': [16, 32],
4     'epochs': [10, 50],
5     'optimizer': ['adam', 'rmsprop'],
6     'model__init': ['he_normal', 'he_uniform']
7 }
8
9 # Configuration de GridSearchCV
10 grid = GridSearchCV(estimator=model, param_grid=param_grid, cv=3)
11
12 # Entraînement du modèle avec recherche des hyperparamètres
13 grid_result = grid.fit(X_train_scaled, y_train)
14
15 # Affichage des meilleurs paramètres et score
16 print(f"Best score: {grid_result.best_score_:.4f}, using Best parameters:
17       {grid_result.best_params_}")

```

Nous obtenons le résultat suivant :

Best score : 0.7824, using Best parameters : 'batch\_size' : 32, 'epochs' : 10, 'model\_\_init' : 'he\_uniform', 'optimizer' : 'rmsprop'

- Entraînement du meilleur modèle

Ce code crée un nouveau modèle Keras qui utilise les meilleurs hyperparamètres trouvés par GridSearchCV. Il entraîne ensuite ce modèle optimisé sur les données d'entraînement, en réservant 20% des données pour la validation.

### Code Python

```
1 # Création d'un modèle Keras avec les meilleurs paramètres
2 best_params = grid_result.best_params_
3 best_model = create_model(optimizer=best_params['optimizer'],
    init=best_params['model_init'])
4
5 # Entraînement du meilleur modèle
6 history = best_model.fit(
7     X_train_scaled, y_train,
8     validation_split=0.2,
9     epochs=best_params['epochs'],
10    batch_size=best_params['batch_size'],
11    verbose=1)
```

- Evaluation et prédictions

Le code qui suit évalue les performances du meilleur modèle sur l'ensemble de test, affichant la perte 'loss' et la précision 'accuracy'. Il utilise ensuite le modèle pour faire des prédictions sur les données de test, convertissant les probabilités en classes binaires (0 ou 1) avec un seuil de 0,5 (point d'équilibre par défaut).

### Code Python

```
1 # Évaluation sur l'ensemble de test
2 test_loss, test_accuracy = best_model.evaluate(X_test_scaled, y_test, verbose=0)
3 print(f"Test loss: {test_loss:.4f}")
4 print(f"Test accuracy: {test_accuracy:.4f}")
5
6 # Prédictions sur l'ensemble de test
7 y_pred = best_model.predict(X_test_scaled)
8 y_pred_class = (y_pred > 0.5).astype(int)
```

Nous obtenons le résultat suivant :

Test loss : 0.4730

Test accuracy : 0.7768

- Métriques de performance

Le code suivant évalue les performances du modèle en affichant un rapport de classification et une matrice de confusion. Également, il calcule et affiche le score AUC-ROC, qui correspond à la mesure de la capacité du modèle à distinguer entre les classes.

### Code Python

```
1 print("\nClassification Report:")
2 print(classification_report(y_test, y_pred_class))
3
4 print("\nConfusion Matrix:")
5 print(confusion_matrix(y_test, y_pred_class))
6
7 # Calculer et afficher le score AUC-ROC
8 roc_auc = roc_auc_score(y_test, y_pred)
9 print(f"\nAUC-ROC Score: {roc_auc:.4f}")
```

Nous obtenons les résultat suivants :

Classification Report :

	precision	recall	f1-score	support
0	0.80	0.73	0.77	17765
1	0.75	0.82	0.79	17764
accuracy			0.78	35529
macro avg	0.78	0.78	0.78	35529
weighted avg	0.78	0.78	0.78	35529

Confusion Matrix :

```
[[ 13006   4759]
 [ 3172  14592]]
```

AUC-ROC Score : 0.8562

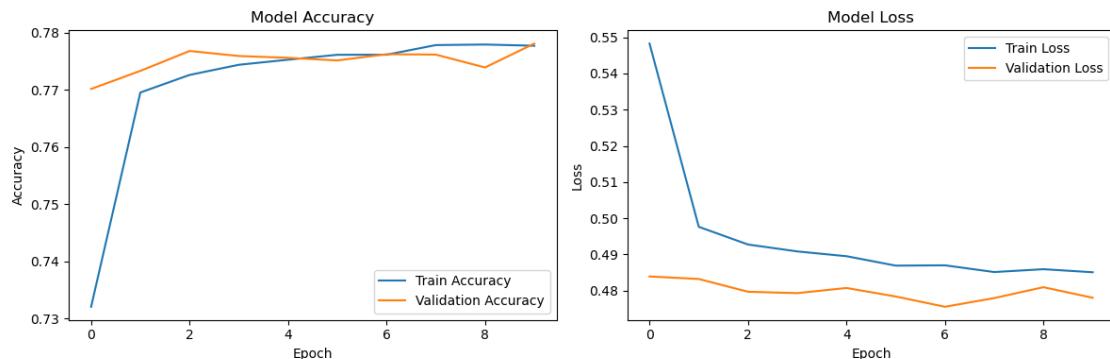
- Tracé de l'historique d'entraînement

Ce code crée un graphique avec deux figures mises côte à côté pour visualiser l'évolution de l'entraînement du modèle. La première figure montre l'évolution de la précision ('accuracy') sur les données d'entraînement et de validation au fil des époques. La seconde figure affiche l'évolution de la perte ('loss') sur ces mêmes ensembles de données.

#### Code Python

```
1 plt.figure(figsize=(12, 4))
2
3 plt.subplot(1, 2, 1)
4 plt.plot(history.history['accuracy'], label='Train Accuracy')
5 plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
6 plt.title('Model Accuracy')
7 plt.xlabel('Epoch')
8 plt.ylabel('Accuracy')
9 plt.legend()
10
11 plt.subplot(1, 2, 2)
12 plt.plot(history.history['loss'], label='Train Loss')
13 plt.plot(history.history['val_loss'], label='Validation Loss')
14 plt.title('Model Loss')
15 plt.xlabel('Epoch')
16 plt.ylabel('Loss')
17 plt.legend()
18
19 plt.tight_layout()
20 plt.show()
```

Nous obtenons les résultats suivants :



- Fonctions d'analyse et de visualisation

- La fonction 'get\_feature\_importance' calcule l'importance des caractéristiques du modèle de réseau de neurones en utilisant les poids de la première couche avec 'weights', puis en calculant la moyenne des valeurs absolues de ces poids avec 'importance'. Elle retourne un dictionnaire associant les noms des caractéristiques à leur importance.
- La fonction 'plot\_feature\_importance' trace un graphique à barres montrant les 20 caractéristiques les plus importantes du modèle (triées par ordre décroissant).
- La fonction 'plot\_roc\_curve' trace la courbe ROC pour évaluer la performance du classificateur binaire. Elle utilise 'roc\_curve' pour calculer le taux de faux positifs (fpr) et le taux de vrais positifs (tpr) en fonction des vraies étiquettes (y\_true) et des probabilités prédictes (y\_pred\_proba). L'AUC est calculée en utilisant auc(fpr, tpr).

Code Python

```

1 def get_feature_importance(model, feature_names):
2     weights = model.layers[0].get_weights()[0]
3     importance = np.mean(np.abs(weights), axis=1)
4     return dict(zip(feature_names, importance))
5
6 def plot_feature_importance(importance_dict, top_n=20):
7     feature_importance = pd.DataFrame({'feature': list(importance_dict.keys()),
8                                         'importance': list(importance_dict.values())})
9     feature_importance = feature_importance.sort_values('importance',
10                                                       ascending=False).head(top_n)
11
12     plt.figure(figsize=(10, 8))
13     sns.barplot(x='importance', y='feature', data=feature_importance)
14     plt.title(f'Top {top_n} des caractéristiques les plus importantes')
15     plt.tight_layout()
16     plt.show()
17
18 def plot_roc_curve(y_true, y_pred_proba):
19     fpr, tpr, _ = roc_curve(y_true, y_pred_proba)
20     roc_auc = auc(fpr, tpr)
21
22     plt.figure(figsize=(8, 6))
23     plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (AUC =
24     {roc_auc:.2f})')
25     plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
26     plt.xlim([0.0, 1.0])
27     plt.ylim([0.0, 1.05])
28     plt.xlabel('Taux de faux positifs')
29     plt.ylabel('Taux de vrais positifs')
30     plt.title('Courbe ROC')
31     plt.legend(loc="lower right")
32     plt.show()

```

- Utilisation des fonctions d'analyse et de visualisation

Ce code appelle la fonction 'get\_feature\_importance' pour pouvoir ensuite générer le graphique issu de la fonction 'plot\_feature\_importance'. Il utilise également les prédictions du modèle sur un ensemble de test pour tracer la courbe ROC via la fonction 'plot\_roc\_curve'.

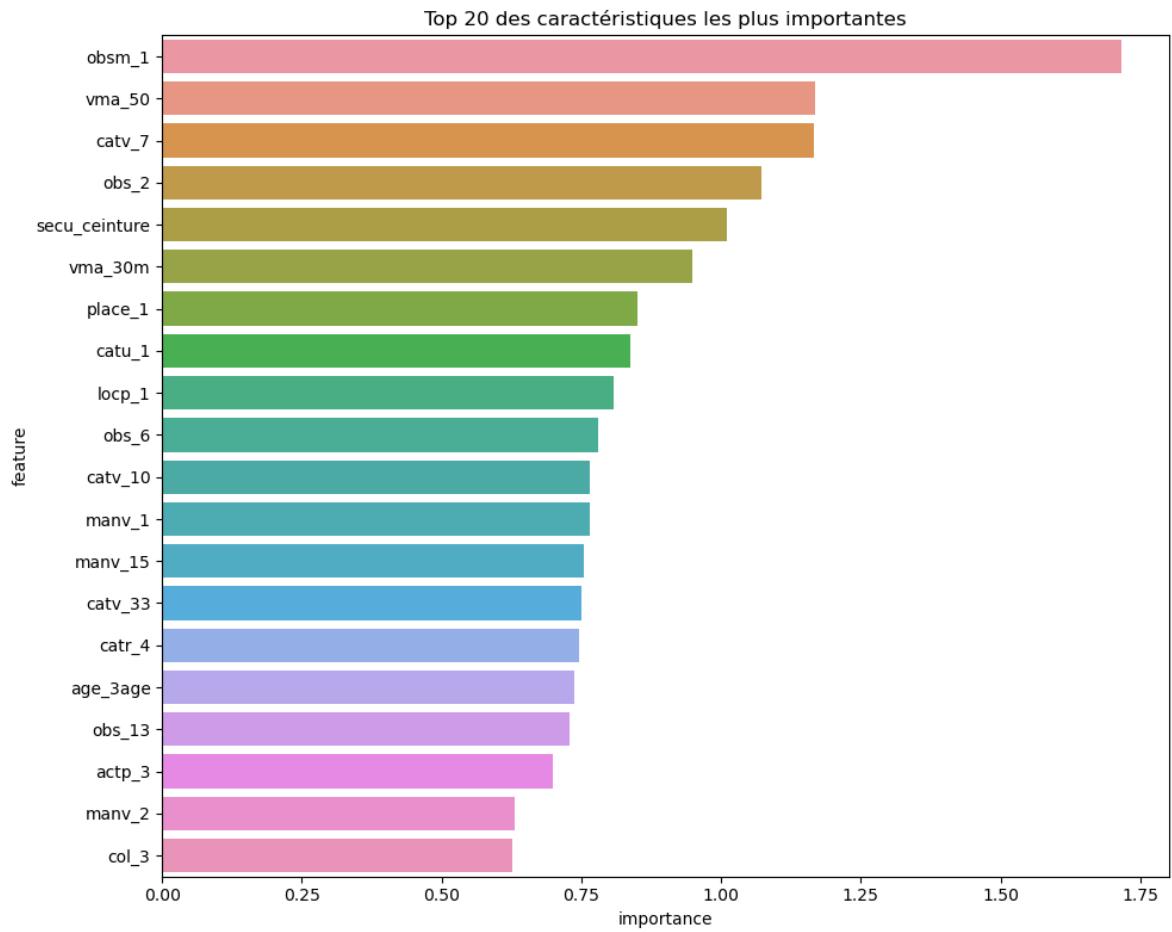
Code Python

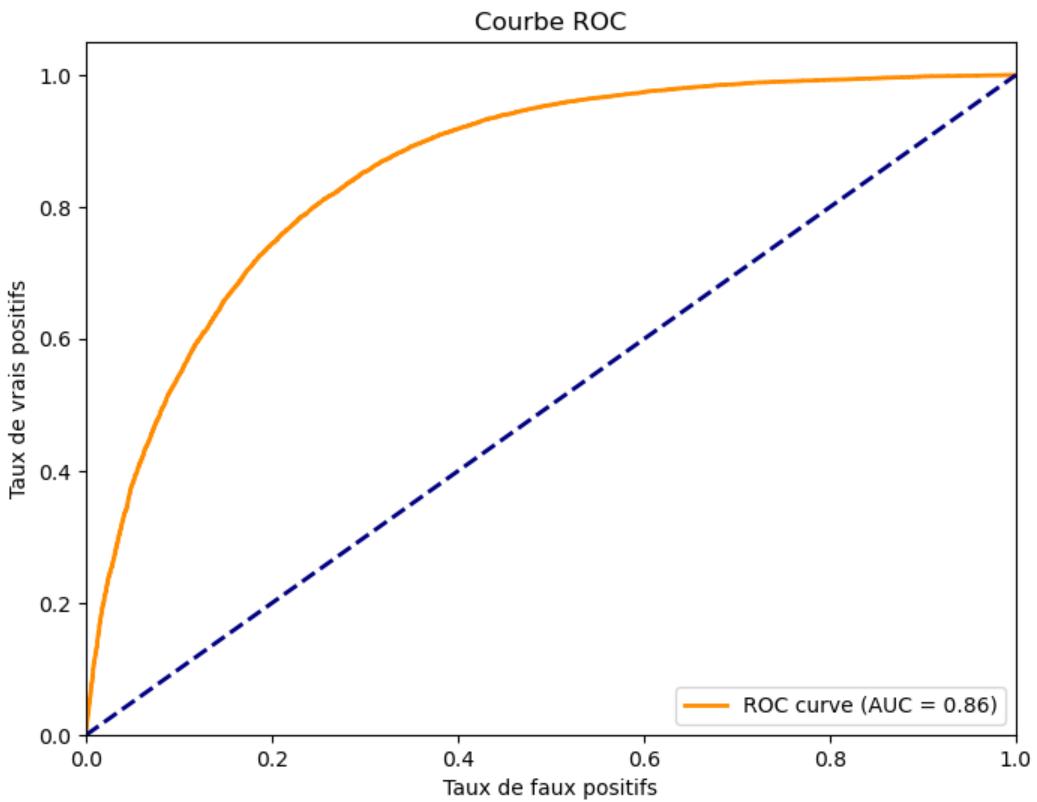
```

1 # Importance des features
2 importance_dict = get_feature_importance(best_model, X.columns)
3 plot_feature_importance(importance_dict)
4
5 # Courbe ROC
6 y_pred_proba = best_model.predict(X_test_scaled).ravel()
7 plot_roc_curve(y_test, y_pred_proba)

```

Nous obtenons les résultats suivants :





- Analyse SHAP

Ce code utilise la bibliothèque SHAP pour expliquer les prédictions du modèle 'best\_model'. Il commence par définir une fonction de prédiction pour le modèle, puis crée un explainer SHAP basé sur un échantillon des données d'entraînement. Ensuite, il calcule les valeurs SHAP pour un sous-ensemble d'échantillons de l'ensemble de test et vérifie les dimensions des résultats. Les valeurs SHAP sont ajustées pour s'assurer qu'elles sont au format approprié, afin de pouvoir générer un graphique récapitulatif visualisant l'importance et l'impact des caractéristiques sur les prédictions du modèle.

#### Code Python

```

1 import shap
2
3 # Définir une fonction de prédiction pour le KernelExplainer
4 def f(X):
5     return best_model.predict(X)
6
7 # Créer un explainer SHAP
8 background = shap.sample(X_train_scaled, 100) # Échantillon de background
9 explainer = shap.KernelExplainer(f, background)
10
11 # Calculer les valeurs SHAP pour un sous-ensemble de l'ensemble de test
12 sample_size = min(100, X_test_scaled.shape[0]) # Prendre au maximum 100 échantillons
13 X_test_sample = X_test_scaled[:sample_size]
14
15 shap_values = explainer.shap_values(X_test_sample)
16
17 # Vérifier les dimensions
18 print("Shape of shap_values:", np.array(shap_values).shape)
19 print("Shape of X_test_sample:", X_test_sample.shape)
20
21 # Ajuster shap_values pour le summary plot

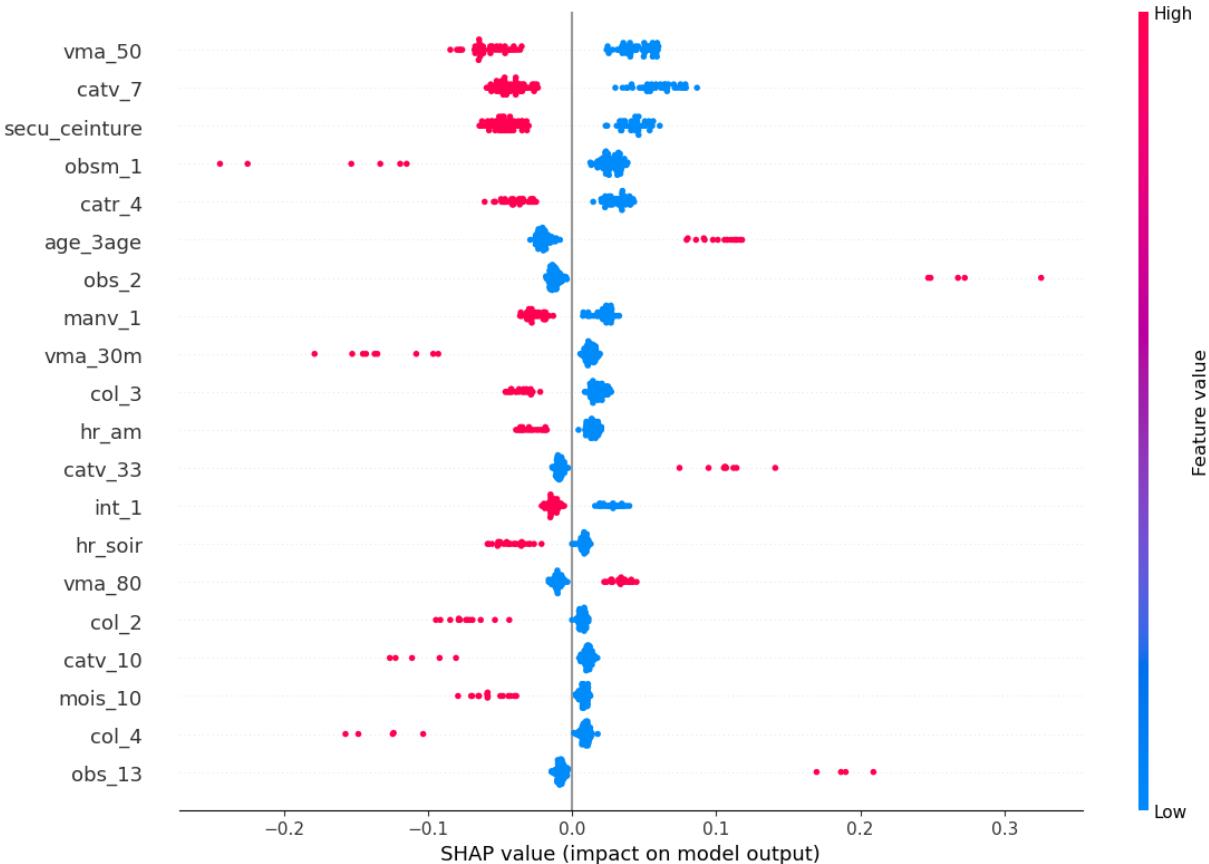
```

```

22 shap_values_adjusted = shap_values[0] if isinstance(shap_values, list) else shap_values
23 shap_values_adjusted = shap_values_adjusted.reshape(shap_values_adjusted.shape[0], -1)
24
25 # Créer un summary plot
26 shap.summary_plot(shap_values_adjusted, X_test_sample, feature_names=X.columns)

```

Nous obtenons le résultat suivant :



## **Annexe 1**

# ANNEXE : ETUDE DES VARIABLES

## 1. Caractéristiques

Rows x columns Rows duplicated

**Caracteristiques** (1176873, 16) 0

### a. Num\_Acc

Description	Numéro d'identifiant de l'accident.																																																																					
Type	int64																																																																					
Etendue des valeurs	<table> <thead> <tr> <th></th><th>count</th><th>unique</th><th>top</th><th>freq</th></tr> </thead> <tbody> <tr> <td><b>Num_Acc</b></td><td>1176873</td><td>1176873</td><td>200500000001</td><td>1</td></tr> </tbody> </table>						count	unique	top	freq	<b>Num_Acc</b>	1176873	1176873	200500000001	1																																																							
	count	unique	top	freq																																																																		
<b>Num_Acc</b>	1176873	1176873	200500000001	1																																																																		
Valeurs nulles	<table> <thead> <tr> <th></th><th>Type</th><th>Val_notnull</th><th>Val_null</th><th>%_null</th></tr> </thead> <tbody> <tr> <td><b>Num_Acc</b></td><td>int64</td><td>1176873</td><td>0</td><td>0.0</td></tr> </tbody> </table>						Type	Val_notnull	Val_null	%_null	<b>Num_Acc</b>	int64	1176873	0	0.0																																																							
	Type	Val_notnull	Val_null	%_null																																																																		
<b>Num_Acc</b>	int64	1176873	0	0.0																																																																		
Outliers	<table> <thead> <tr> <th></th><th>outliers_count</th><th>outliers_unique</th><th>outliers_list</th><th></th></tr> </thead> <tbody> <tr> <td><b>Num_Acc</b></td><td>0</td><td>0</td><td>[ ]</td><td></td></tr> </tbody> </table>						outliers_count	outliers_unique	outliers_list		<b>Num_Acc</b>	0	0	[ ]																																																								
	outliers_count	outliers_unique	outliers_list																																																																			
<b>Num_Acc</b>	0	0	[ ]																																																																			
Répartition	<table> <thead> <tr> <th></th><th>Count</th><th>% valeurs</th><th></th><th></th></tr> </thead> <tbody> <tr> <td><b>Num_Acc</b></td><td></td><td></td><td></td><td></td></tr> <tr> <td><b>200500000001</b></td><td>1</td><td>0.0</td><td></td><td></td></tr> <tr> <td><b>200500000002</b></td><td>1</td><td>0.0</td><td></td><td></td></tr> <tr> <td><b>200500000003</b></td><td>1</td><td>0.0</td><td></td><td></td></tr> <tr> <td><b>200500000004</b></td><td>1</td><td>0.0</td><td></td><td></td></tr> <tr> <td><b>200500000005</b></td><td>1</td><td>0.0</td><td></td><td></td></tr> <tr> <td>...</td><td>...</td><td>...</td><td></td><td></td></tr> <tr> <td><b>202200055298</b></td><td>1</td><td>0.0</td><td></td><td></td></tr> <tr> <td><b>202200055299</b></td><td>1</td><td>0.0</td><td></td><td></td></tr> <tr> <td><b>202200055300</b></td><td>1</td><td>0.0</td><td></td><td></td></tr> <tr> <td><b>202200055301</b></td><td>1</td><td>0.0</td><td></td><td></td></tr> <tr> <td><b>202200055302</b></td><td>1</td><td>0.0</td><td></td><td></td></tr> </tbody> </table> <p>1176873 rows × 2 columns</p>						Count	% valeurs			<b>Num_Acc</b>					<b>200500000001</b>	1	0.0			<b>200500000002</b>	1	0.0			<b>200500000003</b>	1	0.0			<b>200500000004</b>	1	0.0			<b>200500000005</b>	1	0.0			...	...	...			<b>202200055298</b>	1	0.0			<b>202200055299</b>	1	0.0			<b>202200055300</b>	1	0.0			<b>202200055301</b>	1	0.0			<b>202200055302</b>	1	0.0		
	Count	% valeurs																																																																				
<b>Num_Acc</b>																																																																						
<b>200500000001</b>	1	0.0																																																																				
<b>200500000002</b>	1	0.0																																																																				
<b>200500000003</b>	1	0.0																																																																				
<b>200500000004</b>	1	0.0																																																																				
<b>200500000005</b>	1	0.0																																																																				
...	...	...																																																																				
<b>202200055298</b>	1	0.0																																																																				
<b>202200055299</b>	1	0.0																																																																				
<b>202200055300</b>	1	0.0																																																																				
<b>202200055301</b>	1	0.0																																																																				
<b>202200055302</b>	1	0.0																																																																				

<b>Remarque</b>	En 2022, la variable disparait au profit de Accident_Id.
-----------------	--

### b. Accident\_Id

<b>Description</b>	Identifiant de l'accident qui remplace Num_Acc à compter de 2022 (cf. supra)
<b>Remarque</b>	Une modification du nom de colonne est préalablement nécessaire à l'étape de concaténation des dataframes des différentes années.

### c. an

<b>Description</b>	Année de l'accident.			
<b>Type</b>	int64			
<b>Etendue des valeurs</b>	<b>count unique top freq</b> <b>an</b> 1176873 18 5 87026			
<b>Valeurs nulles</b>	<b>Type Val_notnull Val_null %_null</b> <b>an</b> int64 1176873 0 0.0			
<b>Outliers</b>	<b>outliers_count outliers_unique outliers_list</b> <b>an</b> 218404 4 [ 2019.0, 2020.0, 2021.0, 2022.0 ]			

Répartition	Count	% valeurs
<b>an</b>		
<b>5</b>	87026	7.0
<b>6</b>	82993	7.0
<b>7</b>	83850	7.0
<b>8</b>	76767	7.0
<b>9</b>	74409	6.0
<b>10</b>	69379	6.0
<b>11</b>	66974	6.0
<b>12</b>	62250	5.0
<b>13</b>	58397	5.0
<b>14</b>	59854	5.0
<b>15</b>	58654	5.0
<b>16</b>	59432	5.0
<b>17</b>	60701	5.0
<b>18</b>	57783	5.0
<b>2019</b>	58840	5.0
<b>2020</b>	47744	4.0
<b>2021</b>	56518	5.0
<b>2022</b>	55302	5.0
Remarque	Les années ne sont calibrées de la même manière (2 chiffres de 2005 à 2018, 4 chiffres de 2019 à 2022) : +2000 à rajouter aux années de moins de 4 chiffres.	

#### d. mois

Description	Mois de l'accident.										
Type	int64										
Etendue des valeurs	<table> <thead> <tr> <th></th> <th>count</th> <th>unique</th> <th>top</th> <th>freq</th> </tr> </thead> <tbody> <tr> <td><b>mois</b></td> <td>1176873</td> <td>12</td> <td>10</td> <td>111728</td> </tr> </tbody> </table>		count	unique	top	freq	<b>mois</b>	1176873	12	10	111728
	count	unique	top	freq							
<b>mois</b>	1176873	12	10	111728							
Valeurs nulles	<table> <thead> <tr> <th></th> <th>Type</th> <th>Val_notnull</th> <th>Val_null</th> <th>%_null</th> </tr> </thead> <tbody> <tr> <td><b>mois</b></td> <td>int64</td> <td>1176873</td> <td>0</td> <td>0.0</td> </tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>mois</b>	int64	1176873	0	0.0
	Type	Val_notnull	Val_null	%_null							
<b>mois</b>	int64	1176873	0	0.0							

Outliers		outliers_count outliers_unique outliers_list		
	mois	0	0	[ ]
<b>Répartition</b>		<p>mois</p>		
<b>Count % valeurs</b>				
<b>mois</b>				
1	90313	8.0		
2	79959	7.0		
3	90842	8.0		
4	91380	8.0		
5	101060	9.0		
6	111000	9.0		
7	106237	9.0		
8	89006	8.0		
9	109167	9.0		
10	111728	9.0		
11	99941	8.0		
12	96240	8.0		
<b>Evolution</b>		<p>Evolution de la distribution mois</p>		
<b>Remarque</b>		On observe que pendant la période du covid (2020) les proportions changent drastiquement.		

e. jour

Description	Jour de l'accident.
-------------	---------------------

Type	int64										
Etendue des valeurs	<table> <thead> <tr> <th></th><th>count</th><th>unique</th><th>top</th><th>freq</th></tr> </thead> <tbody> <tr> <td>jour</td><td>1176873</td><td>31</td><td>6</td><td>40053</td></tr> </tbody> </table>		count	unique	top	freq	jour	1176873	31	6	40053
	count	unique	top	freq							
jour	1176873	31	6	40053							
Valeurs nulles	<table> <thead> <tr> <th></th><th>Type</th><th>Val_notnull</th><th>Val_null</th><th>%_null</th></tr> </thead> <tbody> <tr> <td>jour</td><td>int64</td><td>1176873</td><td>0</td><td>0.0</td></tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	jour	int64	1176873	0	0.0
	Type	Val_notnull	Val_null	%_null							
jour	int64	1176873	0	0.0							
Outliers	<table> <thead> <tr> <th></th><th>outliers_count</th><th>outliers_unique</th><th>outliers_list</th></tr> </thead> <tbody> <tr> <td>jour</td><td>0</td><td>0</td><td>[ ]</td></tr> </tbody> </table>		outliers_count	outliers_unique	outliers_list	jour	0	0	[ ]		
	outliers_count	outliers_unique	outliers_list								
jour	0	0	[ ]								
Répartition	<p>jour</p>										

	Count	% valeurs
<b>jour</b>		
<b>1</b>	37018	3.0
<b>2</b>	38144	3.0
<b>3</b>	38623	3.0
<b>4</b>	39044	3.0
<b>5</b>	39125	3.0
<b>6</b>	40053	3.0
<b>7</b>	39942	3.0
<b>8</b>	39415	3.0
<b>9</b>	39646	3.0
<b>10</b>	39924	3.0
<b>11</b>	39104	3.0
<b>12</b>	39942	3.0
<b>13</b>	39006	3.0
<b>14</b>	39174	3.0
<b>15</b>	39247	3.0
<b>16</b>	39560	3.0
<b>17</b>	38773	3.0
<b>18</b>	38996	3.0
<b>19</b>	38995	3.0
<b>20</b>	38625	3.0
<b>21</b>	39083	3.0
<b>22</b>	38517	3.0
<b>23</b>	38320	3.0
<b>24</b>	37547	3.0
<b>25</b>	36901	3.0
<b>26</b>	36885	3.0
<b>27</b>	37233	3.0
<b>28</b>	37844	3.0
<b>29</b>	35535	3.0
<b>30</b>	35553	3.0
<b>31</b>	21099	2.0

### f. hrmn

Description	Heures et minutes de l'accident.
Type	[2005-2018] : int64 [2019-2022] : object
Etendue des valeurs	<b>count unique top freq</b> <hr/> <b>hrmn</b> 1176873 2877 1800 14635
Valeurs nulles	<b>Type Val_notnull Val_null %_null</b> <hr/> <b>hrmn</b> object 1176873 0 0.0

Outliers	outliers_count outliers_unique			outliers_list
	hrmn	1123522	2873	[ 00:00, 00:01, 00:02, 00:03, 00:04, 00:05, 00...
Répartition	Count % valeurs			
	hrmn			
	00:00	373	0.0	
	00:01	71	0.0	
	00:02	20	0.0	
	00:03	14	0.0	
	00:04	10	0.0	
	...	...	...	
	955	1581	0.0	
	956	54	0.0	
	957	73	0.0	
	958	77	0.0	
	959	46	0.0	
	2877 rows × 2 columns			
Remarque	Le changement de format des heures et minutes intervenu depuis 2019 nécessitera une uniformisation des valeurs horaires (hh:mm / hhmm / hmm). Après correction, le regroupement par tranches d'une heure sera privilégié.			

## g. lum

Description	Lumière : conditions d'éclairage dans lesquelles l'accident s'est produit :
Modalités	- 1 : Plein jour - 2 : Crénuscuile ou aube - 3 : Nuit sans éclairage public - 4 : Nuit avec éclairage public non allumé - 5 : Nuit avec éclairage public allumé
Type	int64
Etendue des valeurs	count unique top freq
	lum 1176873 6 1 803169
Valeurs nulles	Type Val_notnull Val_null %_null
	lum int64 1176873 0 0.0

Outliers	outliers_count outliers_unique outliers_list																							
	lum	0	0																					
Répartition																								
Count % valeurs	<table border="1"> <thead> <tr> <th>lum</th> <th>Count</th> <th>% valeurs</th> </tr> </thead> <tbody> <tr> <td>-1</td> <td>7</td> <td>0.0</td> </tr> <tr> <td>1</td> <td>803169</td> <td>68.0</td> </tr> <tr> <td>2</td> <td>71424</td> <td>6.0</td> </tr> <tr> <td>3</td> <td>98296</td> <td>8.0</td> </tr> <tr> <td>4</td> <td>9921</td> <td>1.0</td> </tr> <tr> <td>5</td> <td>194056</td> <td>16.0</td> </tr> </tbody> </table>			lum	Count	% valeurs	-1	7	0.0	1	803169	68.0	2	71424	6.0	3	98296	8.0	4	9921	1.0	5	194056	16.0
lum	Count	% valeurs																						
-1	7	0.0																						
1	803169	68.0																						
2	71424	6.0																						
3	98296	8.0																						
4	9921	1.0																						
5	194056	16.0																						
Evolution																								

## h. agg

Description	Localisation agglomération.													
Modalités	<ul style="list-style-type: none"> <li>- 1 : Hors agglomération</li> <li>- 2 : En agglomération</li> </ul>													
Type	int64													
Etendue des valeurs	<table border="1"> <thead> <tr> <th></th> <th>count</th> <th>unique</th> <th>top</th> <th>freq</th> </tr> </thead> <tbody> <tr> <td>agg</td> <td>1176873</td> <td>2</td> <td>2</td> <td>791831</td> </tr> </tbody> </table>					count	unique	top	freq	agg	1176873	2	2	791831
	count	unique	top	freq										
agg	1176873	2	2	791831										
Valeurs nulles	<table border="1"> <thead> <tr> <th></th> <th>Type</th> <th>Val_notnull</th> <th>Val_null</th> <th>%_null</th> </tr> </thead> <tbody> <tr> <td>agg</td> <td>int64</td> <td>1176873</td> <td>0</td> <td>0.0</td> </tr> </tbody> </table>					Type	Val_notnull	Val_null	%_null	agg	int64	1176873	0	0.0
	Type	Val_notnull	Val_null	%_null										
agg	int64	1176873	0	0.0										

Outliers	outliers_count outliers_unique outliers_list
	agg 0 0 []
Répartition	Count % valeurs
agg	<p>1 385042 33.0</p> <p>2 791831 67.0</p>
Evolution	

### i. int

Description	Intersection.										
Modalités	<ul style="list-style-type: none"> <li>- 1 : Hors intersection</li> <li>- 2 : Intersection en X</li> <li>- 3 : Intersection en T</li> <li>- 4 : Intersection en Y</li> <li>- 5 : Intersection à plus de 4 branches</li> <li>- 6 : Giratoire</li> <li>- 7 : Place</li> <li>- 8 : Passage à niveau</li> <li>- 9 : Autre intersection</li> </ul>										
Type	int64										
Etendue des valeurs	<table> <thead> <tr> <th></th> <th>count</th> <th>unique</th> <th>top</th> <th>freq</th> </tr> </thead> <tbody> <tr> <td>int</td> <td>1176873</td> <td>11</td> <td>1</td> <td>820757</td> </tr> </tbody> </table>		count	unique	top	freq	int	1176873	11	1	820757
	count	unique	top	freq							
int	1176873	11	1	820757							

Valeurs nulles	Type	Val_notnull	Val_null	%_null
	int	int64	1176873	0 0.0
Outliers		outliers_count	outliers_unique	outliers_list
	int	103582	7	[ -1.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0 ]
Répartition				
Count % valeurs				
int				
-1 8 0.0				
0 107 0.0				
1 820757 70.0				
2 143573 12.0				
3 108854 9.0				
4 18906 2.0				
5 11463 1.0				
6 35325 3.0				
7 10147 1.0				
8 1486 0.0				
9 26247 2.0				
Evolution				

Years

- 2005
- 2006
- 2007
- 2008
- 2009
- 2010
- 2011
- 2012
- 2013
- 2014
- 2015
- 2016
- 2017
- 2018
- 2019
- 2020
- 2021
- 2022

Modalités

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 1

## j. atm

Description	Conditions atmosphériques.
-------------	----------------------------

Modalités	- -1 : Non renseigné - 1 : Normale - 2 : Pluie légère - 3 : Pluie forte - 4 : Neige - grêle - 5 : Brouillard - fumée - 6 : Vent fort - tempête - 7 : Temps éblouissant - 8 : Temps couvert - 9 : Autre																																							
Type	[2005-2008 ; 2015-2016 ; 2019-2022] : int64 [2009-2014 ; 2017-2018] : float64																																							
Etendue des valeurs	<table> <thead> <tr> <th></th><th>count</th><th>unique</th><th>top</th><th>freq</th></tr> </thead> <tbody> <tr> <td>atm</td><td>1176800</td><td>10</td><td>1.0</td><td>949701</td></tr> </tbody> </table>		count	unique	top	freq	atm	1176800	10	1.0	949701																													
	count	unique	top	freq																																				
atm	1176800	10	1.0	949701																																				
Valeurs nulles	<table> <thead> <tr> <th></th><th>Type</th><th>Val_notnull</th><th>Val_null</th><th>%_null</th></tr> </thead> <tbody> <tr> <td>atm</td><td>float64</td><td>1176800</td><td>73</td><td>0.01</td></tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	atm	float64	1176800	73	0.01																													
	Type	Val_notnull	Val_null	%_null																																				
atm	float64	1176800	73	0.01																																				
Outliers	<table> <thead> <tr> <th></th><th>outliers_count</th><th>outliers_unique</th><th>outliers_list</th></tr> </thead> <tbody> <tr> <td>atm</td><td>227099</td><td>9</td><td>[ -1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0 ]</td></tr> </tbody> </table>		outliers_count	outliers_unique	outliers_list	atm	227099	9	[ -1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0 ]																															
	outliers_count	outliers_unique	outliers_list																																					
atm	227099	9	[ -1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0 ]																																					
Répartition	<table> <thead> <tr> <th></th><th>Count</th><th>% valeurs</th></tr> </thead> <tbody> <tr> <td>atm</td><td></td><td></td></tr> <tr> <td><b>-1.0</b></td><td>22</td><td>0.0</td></tr> <tr> <td><b>1.0</b></td><td>949701</td><td>81.0</td></tr> <tr> <td><b>2.0</b></td><td>123089</td><td>10.0</td></tr> <tr> <td><b>3.0</b></td><td>25270</td><td>2.0</td></tr> <tr> <td><b>4.0</b></td><td>6481</td><td>1.0</td></tr> <tr> <td><b>5.0</b></td><td>7927</td><td>1.0</td></tr> <tr> <td><b>6.0</b></td><td>2933</td><td>0.0</td></tr> <tr> <td><b>7.0</b></td><td>13816</td><td>1.0</td></tr> <tr> <td><b>8.0</b></td><td>39535</td><td>3.0</td></tr> <tr> <td><b>9.0</b></td><td>8026</td><td>1.0</td></tr> <tr> <td><b>Nan</b></td><td>73</td><td>0.0</td></tr> </tbody> </table> <p>The heatmap displays the distribution of weather conditions over time. The x-axis represents the weather condition (Modalites) from 1 to 11, and the y-axis represents the years from 2005 to 2022. The color intensity corresponds to the count of observations for each combination. The most frequent condition is 'Normale' (Modalite 1), which shows a significant increase in frequency from 2005 to 2022. Other conditions like 'Pluie forte' (Modalite 3) and 'Neige - grêle' (Modalite 4) have much lower counts and remain relatively stable.</p>		Count	% valeurs	atm			<b>-1.0</b>	22	0.0	<b>1.0</b>	949701	81.0	<b>2.0</b>	123089	10.0	<b>3.0</b>	25270	2.0	<b>4.0</b>	6481	1.0	<b>5.0</b>	7927	1.0	<b>6.0</b>	2933	0.0	<b>7.0</b>	13816	1.0	<b>8.0</b>	39535	3.0	<b>9.0</b>	8026	1.0	<b>Nan</b>	73	0.0
	Count	% valeurs																																						
atm																																								
<b>-1.0</b>	22	0.0																																						
<b>1.0</b>	949701	81.0																																						
<b>2.0</b>	123089	10.0																																						
<b>3.0</b>	25270	2.0																																						
<b>4.0</b>	6481	1.0																																						
<b>5.0</b>	7927	1.0																																						
<b>6.0</b>	2933	0.0																																						
<b>7.0</b>	13816	1.0																																						
<b>8.0</b>	39535	3.0																																						
<b>9.0</b>	8026	1.0																																						
<b>Nan</b>	73	0.0																																						

<b>Evolution</b>	
<b>Remarque</b>	Les NaN peuvent être remplacés par la valeur -1 qui signifie « non renseigné ».

## k. col

<b>Description</b>	Type de collision.										
<b>Modalités</b>	<ul style="list-style-type: none"> <li>- -1 : Non renseigné</li> <li>- 1 : Deux véhicules - frontale</li> <li>- 2 : Deux véhicules - par l'arrière</li> <li>- 3 : Deux véhicules - par le côté</li> <li>- 4 : Trois véhicules et plus - en chaîne</li> <li>- 5 : Trois véhicules et plus - collisions multiples</li> <li>- 6 : Autre collision</li> <li>- 7 : Sans collision</li> </ul>										
<b>Type</b>	[2005-2009 ; 2012-2015 ; 2019-2022] : int64 [2010-2011 ; 2016-2018] : float64										
<b>Etendue des valeurs</b>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: left;">count</th> <th style="text-align: left;">unique</th> <th style="text-align: left;">top</th> <th style="text-align: left;">freq</th> </tr> </thead> <tbody> <tr> <td><b>col</b></td> <td>1176854</td> <td>8</td> <td>6.0</td> <td>381967</td> </tr> </tbody> </table>		count	unique	top	freq	<b>col</b>	1176854	8	6.0	381967
	count	unique	top	freq							
<b>col</b>	1176854	8	6.0	381967							
<b>Valeurs nulles</b>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: left;">Type</th> <th style="text-align: left;">Val_notnull</th> <th style="text-align: left;">Val_null</th> <th style="text-align: left;">%_null</th> </tr> </thead> <tbody> <tr> <td><b>col</b></td> <td>float64</td> <td>1176854</td> <td>19</td> <td>0.0</td> </tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>col</b>	float64	1176854	19	0.0
	Type	Val_notnull	Val_null	%_null							
<b>col</b>	float64	1176854	19	0.0							
<b>Outliers</b>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: left;">outliers_count</th> <th style="text-align: left;">outliers_unique</th> <th style="text-align: left;">outliers_list</th> </tr> </thead> <tbody> <tr> <td><b>col</b></td> <td>0</td> <td>0</td> <td>[]</td> </tr> </tbody> </table>		outliers_count	outliers_unique	outliers_list	<b>col</b>	0	0	[]		
	outliers_count	outliers_unique	outliers_list								
<b>col</b>	0	0	[]								

<b>Répartition</b>	<table border="1"> <thead> <tr> <th></th><th>Count</th><th>% valeurs</th></tr> </thead> <tbody> <tr> <td><b>col</b></td><td></td><td></td></tr> <tr> <td><b>-1.0</b></td><td>1600</td><td>0.0</td></tr> <tr> <td><b>1.0</b></td><td>115616</td><td>10.0</td></tr> <tr> <td><b>2.0</b></td><td>140178</td><td>12.0</td></tr> <tr> <td><b>3.0</b></td><td>341612</td><td>29.0</td></tr> <tr> <td><b>4.0</b></td><td>37885</td><td>3.0</td></tr> <tr> <td><b>5.0</b></td><td>37244</td><td>3.0</td></tr> <tr> <td><b>6.0</b></td><td>381967</td><td>32.0</td></tr> <tr> <td><b>7.0</b></td><td>120752</td><td>10.0</td></tr> <tr> <td><b>NaN</b></td><td>19</td><td>0.0</td></tr> </tbody> </table> <p>Years: 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022</p>		Count	% valeurs	<b>col</b>			<b>-1.0</b>	1600	0.0	<b>1.0</b>	115616	10.0	<b>2.0</b>	140178	12.0	<b>3.0</b>	341612	29.0	<b>4.0</b>	37885	3.0	<b>5.0</b>	37244	3.0	<b>6.0</b>	381967	32.0	<b>7.0</b>	120752	10.0	<b>NaN</b>	19	0.0
	Count	% valeurs																																
<b>col</b>																																		
<b>-1.0</b>	1600	0.0																																
<b>1.0</b>	115616	10.0																																
<b>2.0</b>	140178	12.0																																
<b>3.0</b>	341612	29.0																																
<b>4.0</b>	37885	3.0																																
<b>5.0</b>	37244	3.0																																
<b>6.0</b>	381967	32.0																																
<b>7.0</b>	120752	10.0																																
<b>NaN</b>	19	0.0																																
<b>Evolution</b>	<p>Modalités: 1, 2, 3, 4, 5, 6, 7, -1</p>																																	
<b>Remarque</b>	<p>Les NaN peuvent être remplacés par la valeur -1 qui signifie « non renseigné ».</p> <p>A noter, une augmentation des modalités non-renseignées en 2020 (covid).</p>																																	

## I. com

<b>Description</b>	Commune : Le numéro de commune est un code donné par l'INSEE. Le code est composé du code INSEE du département suivi par 3 chiffres.										
<b>Type</b>	[2005] : float64 [2006-2018] : int64 [2019-2022] : object										
<b>Etendue des valeurs</b>	<table border="1"> <thead> <tr> <th></th> <th>count</th> <th>unique</th> <th>top</th> <th>freq</th> </tr> </thead> <tbody> <tr> <td><b>com</b></td> <td>1176871</td> <td>23037</td> <td>55</td> <td>33591</td> </tr> </tbody> </table>		count	unique	top	freq	<b>com</b>	1176871	23037	55	33591
	count	unique	top	freq							
<b>com</b>	1176871	23037	55	33591							

Valeurs nulles	Type	Val_notnull	Val_null	%_null
	com	object	1176871	2 0.0
Outliers	outliers_count	outliers_unique		outliers_list
	com	1130160	23036	[ 0.0, 01001, 01004, 01005, 01007, 01008, 0101...
	<p style="text-align: center;">Boxplots pour: com</p>			
Répartition	Count	% valeurs		
	com			
	0.0	1	0.0	
	01001	4	0.0	
	01004	26	0.0	
	01005	3	0.0	
	01007	7	0.0	
	...	...	...	
	98833	3	0.0	
	99	1818	0.0	
	99.0	200	0.0	
	N/C	1	0.0	
	NaN	2	0.0	
	23038 rows × 2 columns			
Remarque	Les codes « communes » ne sont pas enregistrés sous le même format, et présence de valeurs non pertinentes telles que N/C, 0 ou NaN (voir si le code département permet sa reconstitution, puisqu'à partir de 2019, la variable change d'aspect pour répondre à celui de la description).			

## m.adr

Description	Adresse postale : variable renseignée pour les accidents survenus en agglomération.				
Type	object				
Etendue des valeurs	<b>count unique top freq</b> <b>adr</b> 1032364 483819 AUTOROUTE A86 4268				
Valeurs nulles	<b>Type Val_notnull Val_null %_null</b> <b>adr</b> object 1032364 144509 12.28				
Outliers	<b>outliers_count outliers_unique outliers_list</b> <b>adr</b> 1032364 483819 [ A64, (Bd FELIX mercarder, (CAMP MAJOR), ...				
Répartition	<b>Count % valeurs</b> <b>adr</b> <b>A64</b> 1 0.0 <b>(Bd FELIX mercarder</b> 1 0.0 <b>(CAMP MAJOR)</b> 1 0.0 <b>(ROUTE DE DIEPPE)</b> 2 0.0 <b>(nouvelle rocade)</b> 1 0.0 <b>...</b> ... ... <b>île HTR DU PALAIS DU MAROC</b> 1 0.0 <b>île hauteur de Vilormel</b> 1 0.0 <b>île proximité RD1075/50A</b> 1 0.0 <b>ôté droit dans le sens</b> 1 0.0 <b>NaN</b> 144509 12.0				
Remarque	Cette variable s'avère non pertinente en raison de sa dispersion.				

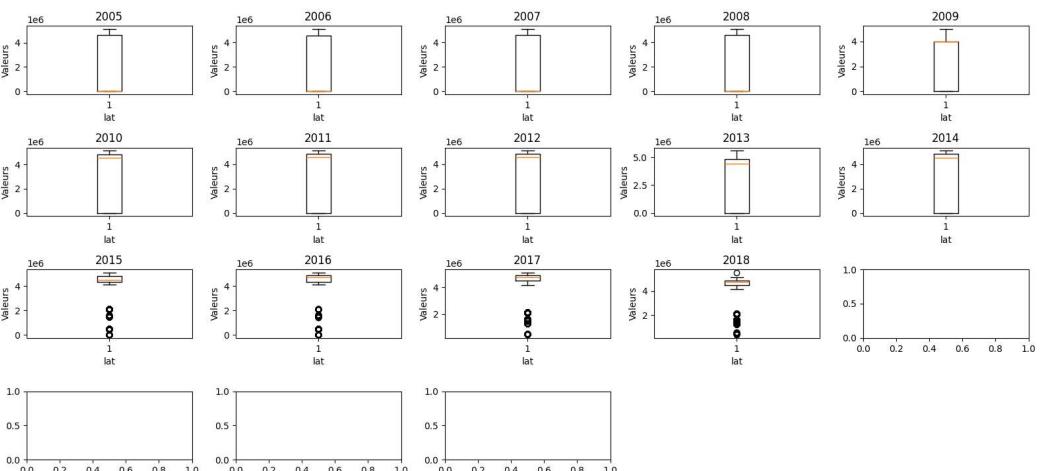
## n.gps

Description	Codage GPS : 1 caractère indicateur de provenance : M = Métropole A = Antilles (Martinique ou Guadeloupe) G = Guyane R = Réunion Y = Mayotte
-------------	---

Type	object																																				
Etendue des valeurs	<b>count unique top freq</b> <b>gps</b> 480052 10 M 462639																																				
Valeurs nulles	<b>Type Val_notnull Val_null %_null</b> <b>gps</b> object 480052 696821 59.21																																				
Outliers	<b>outliers_count outliers_unique outliers_list</b> <b>gps</b> 17413 9 [ 0, A, C, G, P, R, S, T, Y ]																																				
Répartition	<p><b>Count % valeurs</b></p> <table> <thead> <tr> <th>gps</th> <th>Count</th> <th>% valeurs</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>9</td> <td>0.0</td> </tr> <tr> <td>A</td> <td>7850</td> <td>1.0</td> </tr> <tr> <td>C</td> <td>2</td> <td>0.0</td> </tr> <tr> <td>G</td> <td>3387</td> <td>0.0</td> </tr> <tr> <td>M</td> <td>462639</td> <td>39.0</td> </tr> <tr> <td>P</td> <td>1</td> <td>0.0</td> </tr> <tr> <td>R</td> <td>5316</td> <td>0.0</td> </tr> <tr> <td>S</td> <td>4</td> <td>0.0</td> </tr> <tr> <td>T</td> <td>2</td> <td>0.0</td> </tr> <tr> <td>Y</td> <td>842</td> <td>0.0</td> </tr> <tr> <td><b>Nan</b></td> <td>696821</td> <td>59.0</td> </tr> </tbody> </table> <p>Count</p> <p>Modalités</p> <p>Years</p> <ul style="list-style-type: none"> <li>2005</li> <li>2006</li> <li>2007</li> <li>2008</li> <li>2009</li> <li>2010</li> <li>2011</li> <li>2012</li> <li>2013</li> <li>2014</li> <li>2015</li> <li>2016</li> <li>2017</li> <li>2018</li> </ul>	gps	Count	% valeurs	0	9	0.0	A	7850	1.0	C	2	0.0	G	3387	0.0	M	462639	39.0	P	1	0.0	R	5316	0.0	S	4	0.0	T	2	0.0	Y	842	0.0	<b>Nan</b>	696821	59.0
gps	Count	% valeurs																																			
0	9	0.0																																			
A	7850	1.0																																			
C	2	0.0																																			
G	3387	0.0																																			
M	462639	39.0																																			
P	1	0.0																																			
R	5316	0.0																																			
S	4	0.0																																			
T	2	0.0																																			
Y	842	0.0																																			
<b>Nan</b>	696821	59.0																																			
Evolution	<p>Evolution de la distribution gps</p> <p>Proportion (%)</p> <p>Années</p> <p>Modalités</p> <ul style="list-style-type: none"> <li>S</li> <li>C</li> <li>G</li> <li>P</li> <li>A</li> <li>M</li> <li>O</li> <li>T</li> <li>R</li> <li>Y</li> </ul>																																				

<b>Remarque</b>	Cette variable présente un grand nombre de valeurs NaN, notamment depuis sa disparition en 2019, ce qui peut la rendre non pertinente.
-----------------	--

## o. lat

<b>Description</b>	Latitude.				
<b>Type</b>	[2005-2018] : float64 [2019-2022] : object				
<b>Etendue des valeurs</b>	<b>count unique top freq</b> <b>lat</b> 689805 379834 0.0 117839				
<b>Valeurs nulles</b>	<b>Type Val_notnull Val_null %_null</b> <b>lat</b> object 689805 487068 41.39				
<b>Outliers</b>	<b>outliers_count outliers_unique</b> <b>lat</b> 549485 379832 [-12,6853290000, -12,6894530, -12,69219500...  Boxplots pour: lat				
					

Répartition	Count % valeurs		
	lat		
	-12,6853290000	1	0.0
	-12,6894530	1	0.0
	-12,6921950000	1	0.0
	-12,7031220000	1	0.0
	-12,7044830	1	0.0
	...	...	...
	942686.0	1	0.0
	944429.0	1	0.0
379835 rows × 2 columns			
Remarque	Le format des coordonnées ne semble pas uniforme avec un taux élevé de valeurs NaN.		

### p. long

Description	Longitude			
Type	[2005-2008 ; 2010-2018] : float64 [2009 ; 2019-2022] : object			
Etendue des valeurs	<b>count unique top freq</b> <b>long</b> 689801 415250 0.0 107376			
Valeurs nulles	<b>Type Val_notnull Val_null %_null</b> <b>long</b> object 689801 487072 41.39			
Outliers	<b>outliers_count outliers_unique outliers_list</b> <b>long</b> 552144 415248 [-0,0003420000, -0,0004390, -0,0005150000,...]			

Répartition		Count	% valeurs
<b>long</b>			
	<b>-0,0003420000</b>	1	0.0
	<b>-0,0004390</b>	1	0.0
	<b>-0,0005150000</b>	1	0.0
	<b>-0,0006440000</b>	1	0.0
	<b>-0,0012150</b>	1	0.0
	...	...	...
	<b>9998.0</b>	1	0.0
	<b>99980.0</b>	1	0.0
	<b>99984.0</b>	1	0.0
	<b>99999.0</b>	2	0.0
	<b>NaN</b>	487072	41.0
415251 rows × 2 columns			
Remarque	Le format des coordonnées ne semblent pas uniformes avec un taux élevé de valeurs NaN.		

### q. dep

Description	Département : Code INSEE du département (2A Corse-du-Sud – 2B Haute-Corse).										
Type	[2005-2018] : int64 [2019-2022] : object										
Etendue des valeurs	<table> <thead> <tr> <th></th> <th>count</th> <th>unique</th> <th>top</th> <th>freq</th> </tr> </thead> <tbody> <tr> <td><b>dep</b></td> <td>1176873</td> <td>204</td> <td>750</td> <td>99021</td> </tr> </tbody> </table>		count	unique	top	freq	<b>dep</b>	1176873	204	750	99021
	count	unique	top	freq							
<b>dep</b>	1176873	204	750	99021							
Valeurs nulles	<table> <thead> <tr> <th></th> <th>Type</th> <th>Val_notnull</th> <th>Val_null</th> <th>%_null</th> </tr> </thead> <tbody> <tr> <td><b>dep</b></td> <td>object</td> <td>1176873</td> <td>0</td> <td>0.0</td> </tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>dep</b>	object	1176873	0	0.0
	Type	Val_notnull	Val_null	%_null							
<b>dep</b>	object	1176873	0	0.0							
Outliers	<table> <thead> <tr> <th></th> <th>outliers_count</th> <th>outliers_unique</th> <th>outliers_list</th> </tr> </thead> <tbody> <tr> <td><b>dep</b></td> <td>575077</td> <td>180</td> <td>[ 01, 02, 03, 04, 05, 06, 07, 08, 09, 1, 10, 1...</td> </tr> </tbody> </table>		outliers_count	outliers_unique	outliers_list	<b>dep</b>	575077	180	[ 01, 02, 03, 04, 05, 06, 07, 08, 09, 1, 10, 1...		
	outliers_count	outliers_unique	outliers_list								
<b>dep</b>	575077	180	[ 01, 02, 03, 04, 05, 06, 07, 08, 09, 1, 10, 1...								

Répartition	Count % valeurs	
	dep	
<b>01</b>	1291	0.0
<b>02</b>	625	0.0
<b>03</b>	694	0.0
<b>04</b>	605	0.0
<b>05</b>	700	0.0
...	...	...
<b>977</b>	58	0.0
<b>978</b>	125	0.0
<b>986</b>	46	0.0
<b>987</b>	539	0.0
<b>988</b>	1220	0.0

204 rows × 2 columns

## 2. Lieux

Rows x columns Rows duplicated

Lieux (1176873, 19) 0

### a. Num\_Acc

Description	Identifiant de l'accident identique à celui du fichier "rubrique CARACTERISTIQUES" repris dans l'accident.														
Type	int64														
Etendue des valeurs	<table> <thead> <tr> <th></th><th>count</th><th>unique</th><th>top</th><th>freq</th></tr> </thead> <tbody> <tr> <td>Num_Acc</td><td>1176873</td><td>1176873</td><td>200500000001</td><td>1</td></tr> </tbody> </table>						count	unique	top	freq	Num_Acc	1176873	1176873	200500000001	1
	count	unique	top	freq											
Num_Acc	1176873	1176873	200500000001	1											
Valeurs nulles	<table> <thead> <tr> <th></th><th>Type</th><th>Val_notnull</th><th>Val_null</th><th>%_null</th></tr> </thead> <tbody> <tr> <td>Num_Acc</td><td>int64</td><td>1176873</td><td>0</td><td>0.0</td></tr> </tbody> </table>						Type	Val_notnull	Val_null	%_null	Num_Acc	int64	1176873	0	0.0
	Type	Val_notnull	Val_null	%_null											
Num_Acc	int64	1176873	0	0.0											
Outliers	<table> <thead> <tr> <th></th><th>outliers_count</th><th>outliers_unique</th><th>outliers_list</th></tr> </thead> <tbody> <tr> <td>Num_Acc</td><td>0</td><td>0</td><td>[]</td></tr> </tbody> </table>						outliers_count	outliers_unique	outliers_list	Num_Acc	0	0	[]		
	outliers_count	outliers_unique	outliers_list												
Num_Acc	0	0	[]												

Répartition	Count % valeurs		
	Num_Acc		
200500000001	1	0.0	
200500000002	1	0.0	
200500000003	1	0.0	
200500000004	1	0.0	
200500000005	1	0.0	
...	...	...	
202200055298	1	0.0	
202200055299	1	0.0	
202200055300	1	0.0	
202200055301	1	0.0	
202200055302	1	0.0	
1176873 rows × 2 columns			

## b. catr

Description	Catégorie de route.										
Modalités	<ul style="list-style-type: none"> <li>- 1 : Autoroute</li> <li>- 2 : Route nationale</li> <li>- 3 : Route Départementale</li> <li>- 4 : Voie Communale</li> <li>- 5 : Hors réseau public</li> <li>- 6 : Parc de stationnement ouvert à la circulation publique</li> <li>- 7 : Routes de métropole urbaine</li> <li>- 9 : Autre</li> </ul>										
Type	[2005] : float64 [2006-2022] : int64										
Etendue des valeurs	<table> <thead> <tr> <th></th> <th>count</th> <th>unique</th> <th>top</th> <th>freq</th> </tr> </thead> <tbody> <tr> <td>catr</td> <td>1176872</td> <td>8</td> <td>4.0</td> <td>571872</td> </tr> </tbody> </table>		count	unique	top	freq	catr	1176872	8	4.0	571872
	count	unique	top	freq							
catr	1176872	8	4.0	571872							
Valeurs nulles	<table> <thead> <tr> <th></th> <th>Type</th> <th>Val_notnull</th> <th>Val_null</th> <th>%_null</th> </tr> </thead> <tbody> <tr> <td>catr</td> <td>float64</td> <td>1176872</td> <td>1</td> <td>0.0</td> </tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	catr	float64	1176872	1	0.0
	Type	Val_notnull	Val_null	%_null							
catr	float64	1176872	1	0.0							
Outliers	<table> <thead> <tr> <th></th> <th>outliers_count</th> <th>outliers_unique</th> <th>outliers_list</th> </tr> </thead> <tbody> <tr> <td>catr</td> <td>127334</td> <td>4</td> <td>[ 1.0, 6.0, 7.0, 9.0 ]</td> </tr> </tbody> </table>		outliers_count	outliers_unique	outliers_list	catr	127334	4	[ 1.0, 6.0, 7.0, 9.0 ]		
	outliers_count	outliers_unique	outliers_list								
catr	127334	4	[ 1.0, 6.0, 7.0, 9.0 ]								

<b>Répartition</b>	<table border="1"> <thead> <tr> <th></th><th>Count</th><th>% valeurs</th></tr> </thead> <tbody> <tr> <td><b>catr</b></td><td></td><td></td></tr> <tr> <td><b>1.0</b></td><td>93089</td><td>8.0</td></tr> <tr> <td><b>2.0</b></td><td>89780</td><td>8.0</td></tr> <tr> <td><b>3.0</b></td><td>385809</td><td>33.0</td></tr> <tr> <td><b>4.0</b></td><td>571872</td><td>49.0</td></tr> <tr> <td><b>5.0</b></td><td>2077</td><td>0.0</td></tr> <tr> <td><b>6.0</b></td><td>7982</td><td>1.0</td></tr> <tr> <td><b>7.0</b></td><td>7275</td><td>1.0</td></tr> <tr> <td><b>9.0</b></td><td>18988</td><td>2.0</td></tr> <tr> <td><b>NaN</b></td><td>1</td><td>0.0</td></tr> </tbody> </table>		Count	% valeurs	<b>catr</b>			<b>1.0</b>	93089	8.0	<b>2.0</b>	89780	8.0	<b>3.0</b>	385809	33.0	<b>4.0</b>	571872	49.0	<b>5.0</b>	2077	0.0	<b>6.0</b>	7982	1.0	<b>7.0</b>	7275	1.0	<b>9.0</b>	18988	2.0	<b>NaN</b>	1	0.0
	Count	% valeurs																																
<b>catr</b>																																		
<b>1.0</b>	93089	8.0																																
<b>2.0</b>	89780	8.0																																
<b>3.0</b>	385809	33.0																																
<b>4.0</b>	571872	49.0																																
<b>5.0</b>	2077	0.0																																
<b>6.0</b>	7982	1.0																																
<b>7.0</b>	7275	1.0																																
<b>9.0</b>	18988	2.0																																
<b>NaN</b>	1	0.0																																
<b>Evolution</b>																																		
<b>Remarque</b>	La valeur NaN peut être supprimée.																																	

### c. voie

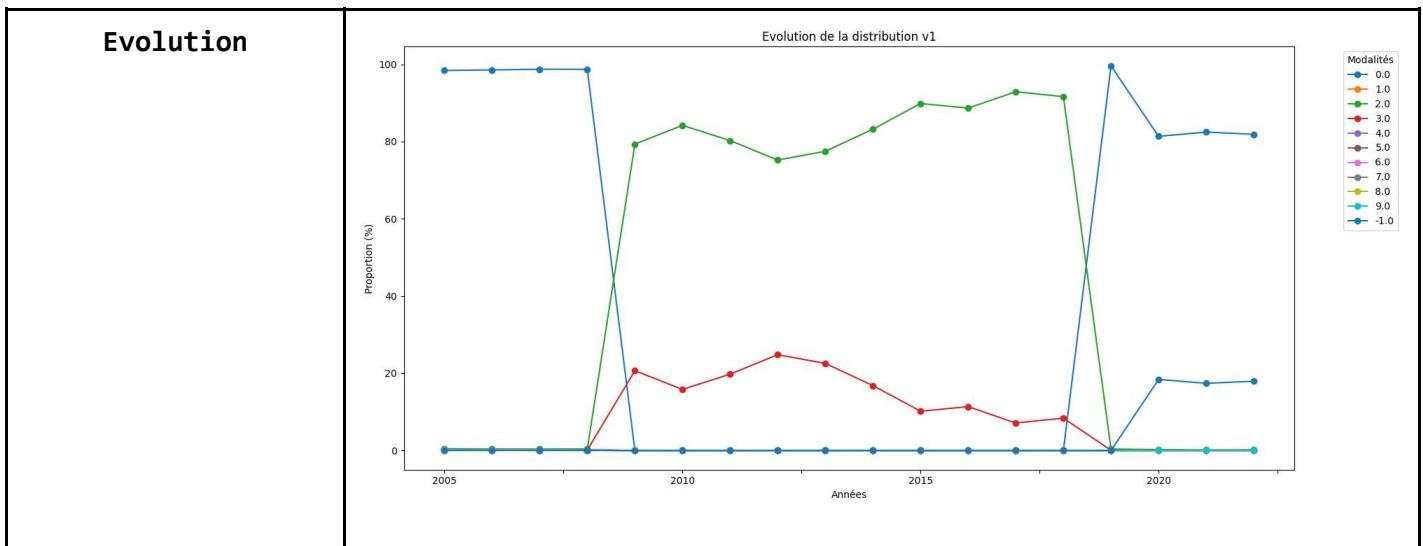
<b>Description</b>	Numéro de la route.										
<b>Type</b>	[2005-2015] : float64 [2016-2022] : object										
<b>Etendue des valeurs</b>	<table> <thead> <tr> <th></th> <th>count</th> <th>unique</th> <th>top</th> <th>freq</th> </tr> </thead> <tbody> <tr> <td><b>voie</b></td> <td>1064888</td> <td>38865</td> <td>0.0</td> <td>429071</td> </tr> </tbody> </table>		count	unique	top	freq	<b>voie</b>	1064888	38865	0.0	429071
	count	unique	top	freq							
<b>voie</b>	1064888	38865	0.0	429071							
<b>Valeurs nulles</b>	<table> <thead> <tr> <th></th> <th>Type</th> <th>Val_notnull</th> <th>Val_null</th> <th>%_null</th> </tr> </thead> <tbody> <tr> <td><b>voie</b></td> <td>object</td> <td>1064888</td> <td>111985</td> <td>9.52</td> </tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>voie</b>	object	1064888	111985	9.52
	Type	Val_notnull	Val_null	%_null							
<b>voie</b>	object	1064888	111985	9.52							

Outliers	<p><b>outliers_count outliers_unique outliers_list</b></p> <p><b>voie</b> 563558 38862 [ (R), ...</p> <p>Boxplots pour: voie</p>																																				
Répartition	<p><b>Count % valeurs</b></p> <table border="1"> <thead> <tr> <th>voie</th> <th>Count</th> <th>% valeurs</th> </tr> </thead> <tbody> <tr> <td>(R)</td> <td>1</td> <td>0.0</td> </tr> <tr> <td>(AV)</td> <td>1</td> <td>0.0</td> </tr> <tr> <td>(AV)</td> <td>1</td> <td>0.0</td> </tr> <tr> <td>(AV)</td> <td>1</td> <td>0.0</td> </tr> <tr> <td>(BD)</td> <td>1</td> <td>0.0</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>x</td> <td>3</td> <td>0.0</td> </tr> <tr> <td>xxxx</td> <td>1</td> <td>0.0</td> </tr> <tr> <td><b>ÉCHANGEUR DU RONDEAU</b></td> <td>2</td> <td>0.0</td> </tr> <tr> <td><b>Épalle</b></td> <td>1</td> <td>0.0</td> </tr> <tr> <td><b>Nan</b></td> <td>111985</td> <td>10.0</td> </tr> </tbody> </table> <p>38866 rows × 2 columns</p>	voie	Count	% valeurs	(R)	1	0.0	(AV)	1	0.0	(AV)	1	0.0	(AV)	1	0.0	(BD)	1	0.0	...	...	...	x	3	0.0	xxxx	1	0.0	<b>ÉCHANGEUR DU RONDEAU</b>	2	0.0	<b>Épalle</b>	1	0.0	<b>Nan</b>	111985	10.0
voie	Count	% valeurs																																			
(R)	1	0.0																																			
(AV)	1	0.0																																			
(AV)	1	0.0																																			
(AV)	1	0.0																																			
(BD)	1	0.0																																			
...	...	...																																			
x	3	0.0																																			
xxxx	1	0.0																																			
<b>ÉCHANGEUR DU RONDEAU</b>	2	0.0																																			
<b>Épalle</b>	1	0.0																																			
<b>Nan</b>	111985	10.0																																			
Remarque	Les valeurs NaN sont élevées, ce qui peut rendre la variable non pertinente.																																				

### d. v1

<b>Description</b>	Indice numérique du numéro de route (exemple : 2 bis, 3 ter etc.).
--------------------	--

Type	[2020-2022] : int64 [2005-2019] : float64																										
Etendue des valeurs	<b>count unique top freq</b> <b>v1</b> 541049 11 0.0 504288																										
Valeurs nulles	<b>Type Val_notnull Val_null %_null</b> <b>v1</b> float64 541049 635824 54.03																										
Outliers	<b>outliers_count outliers_unique outliers_list</b> <b>v1</b> 36761 10 [-1.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0...]																										
Répartition	<table> <thead> <tr> <th>Count % valeurs</th> <th>v1</th> </tr> </thead> <tbody> <tr> <td>-1.0 28529 2.0</td> <td></td> </tr> <tr> <td>0.0 504288 43.0</td> <td></td> </tr> <tr> <td>1.0 1020 0.0</td> <td></td> </tr> <tr> <td>2.0 4302 0.0</td> <td></td> </tr> <tr> <td>3.0 887 0.0</td> <td></td> </tr> <tr> <td>4.0 396 0.0</td> <td></td> </tr> <tr> <td>5.0 232 0.0</td> <td></td> </tr> <tr> <td>6.0 461 0.0</td> <td></td> </tr> <tr> <td>7.0 392 0.0</td> <td></td> </tr> <tr> <td>8.0 247 0.0</td> <td></td> </tr> <tr> <td>9.0 295 0.0</td> <td></td> </tr> <tr> <td><b>Nan</b> 635824 54.0</td> <td></td> </tr> </tbody> </table> <p>The figure is a stacked bar chart titled 'v1'. The x-axis is labeled 'Modalités' and has ticks at 1.0, 2.0, 3.0, 4.0, 5.0, 7.0, 8.0, 9.0, and -1.0. The y-axis is labeled 'Count' and ranges from 0 to 500,000. There are two main bars: one for 'NaN' reaching nearly 600,000 and one for '0.0' reaching approximately 300,000. Each bar is composed of multiple colored segments representing different years from 2005 to 2022. The colors transition from dark purple for earlier years to bright yellow for later years. The '0.0' bar shows a clear peak for the year 2006.</p>	Count % valeurs	v1	-1.0 28529 2.0		0.0 504288 43.0		1.0 1020 0.0		2.0 4302 0.0		3.0 887 0.0		4.0 396 0.0		5.0 232 0.0		6.0 461 0.0		7.0 392 0.0		8.0 247 0.0		9.0 295 0.0		<b>Nan</b> 635824 54.0	
Count % valeurs	v1																										
-1.0 28529 2.0																											
0.0 504288 43.0																											
1.0 1020 0.0																											
2.0 4302 0.0																											
3.0 887 0.0																											
4.0 396 0.0																											
5.0 232 0.0																											
6.0 461 0.0																											
7.0 392 0.0																											
8.0 247 0.0																											
9.0 295 0.0																											
<b>Nan</b> 635824 54.0																											



### e. v2

Description	Lettre indice alphanumérique de la route.
Type	object
Etendue des valeurs	<b>count unique top freq</b> <b>v2</b> 56624 74 A 24722
Valeurs nulles	<b>Type Val_notnull Val_null %_null</b> <b>v2</b> object 56624 1120249 95.19
Outliers	<b>outliers_count outliers_unique outliers_list</b> <b>v2</b> 18298 72 [□, -, D, ., 0, 1, 15, 1A, 2, 3, 34, 4, 5, ...

Répartition	Count % valeurs		
<b>v2</b>			
	□	537	0.0
	-	246	0.0
	D	1	0.0
	.	1	0.0
	0	1099	0.0
	...	...	...
	v	2	0.0
	w	2	0.0
	y	3	0.0
	z	19	0.0
	<b>NaN</b>	1120249	95.0
75 rows × 2 columns			
Remarque	Les informations rassemblées sont parfois de type incohérent, d'autant plus qu'il y a un grand nombre de valeurs NaN.		

## f. circ

Description	Régime de circulation										
Modalités	<ul style="list-style-type: none"> <li>- -1 : Non renseigné</li> <li>- 1 : A sens unique</li> <li>- 2 : Bidirectionnelle</li> <li>- 3 : A chaussées séparées</li> <li>- 4 : Avec voies d'affectation variable</li> </ul>										
Type	int64										
Etendue des valeurs	<table> <thead> <tr> <th></th> <th>count</th> <th>unique</th> <th>top</th> <th>freq</th> </tr> </thead> <tbody> <tr> <td><b>circ</b></td><td>1175299</td><td>6</td><td>2.0</td><td>741823</td></tr> </tbody> </table>		count	unique	top	freq	<b>circ</b>	1175299	6	2.0	741823
	count	unique	top	freq							
<b>circ</b>	1175299	6	2.0	741823							
Valeurs nulles	<table> <thead> <tr> <th></th> <th>Type</th> <th>Val_notnull</th> <th>Val_null</th> <th>%_null</th> </tr> </thead> <tbody> <tr> <td><b>circ</b></td><td>float64</td><td>1175299</td><td>1574</td><td>0.13</td></tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>circ</b>	float64	1175299	1574	0.13
	Type	Val_notnull	Val_null	%_null							
<b>circ</b>	float64	1175299	1574	0.13							
Outliers	<table> <thead> <tr> <th></th> <th>outliers_count</th> <th>outliers_unique</th> <th>outliers_list</th> </tr> </thead> <tbody> <tr> <td><b>circ</b></td><td>433476</td><td>5</td><td>[ -1.0, 0.0, 1.0, 3.0, 4.0 ]</td></tr> </tbody> </table>		outliers_count	outliers_unique	outliers_list	<b>circ</b>	433476	5	[ -1.0, 0.0, 1.0, 3.0, 4.0 ]		
	outliers_count	outliers_unique	outliers_list								
<b>circ</b>	433476	5	[ -1.0, 0.0, 1.0, 3.0, 4.0 ]								

<b>Répartition</b>	<p><b>Count % valeurs</b></p> <table border="1"> <thead> <tr> <th>circ</th><th>Count</th><th>% valeurs</th></tr> </thead> <tbody> <tr> <td>-1.0</td><td>12184</td><td>1.0</td></tr> <tr> <td>0.0</td><td>49966</td><td>4.0</td></tr> <tr> <td>1.0</td><td>208883</td><td>18.0</td></tr> <tr> <td>2.0</td><td>741823</td><td>63.0</td></tr> <tr> <td>3.0</td><td>155773</td><td>13.0</td></tr> <tr> <td>4.0</td><td>6670</td><td>1.0</td></tr> <tr> <td>NaN</td><td>1574</td><td>0.0</td></tr> </tbody> </table>	circ	Count	% valeurs	-1.0	12184	1.0	0.0	49966	4.0	1.0	208883	18.0	2.0	741823	63.0	3.0	155773	13.0	4.0	6670	1.0	NaN	1574	0.0
circ	Count	% valeurs																							
-1.0	12184	1.0																							
0.0	49966	4.0																							
1.0	208883	18.0																							
2.0	741823	63.0																							
3.0	155773	13.0																							
4.0	6670	1.0																							
NaN	1574	0.0																							
<b>Evolution</b>																									
<b>Remarque</b>	Les NaN peuvent être remplacés par la valeur -1 qui signifie « non renseigné ».																								

### g. nbv

<b>Description</b>	Nombre total de voies de circulation.										
<b>Type</b>	[2005-2008 ; 2019-2021] : int64 [2009-2018] : float64 [2022] : object										
<b>Etendue des valeurs</b>	<table> <thead> <tr> <th></th> <th>count</th> <th>unique</th> <th>top</th> <th>freq</th> </tr> </thead> <tbody> <tr> <td><b>nbv</b></td> <td>1174142</td> <td>70</td> <td>2.0</td> <td>351510</td> </tr> </tbody> </table>		count	unique	top	freq	<b>nbv</b>	1174142	70	2.0	351510
	count	unique	top	freq							
<b>nbv</b>	1174142	70	2.0	351510							
<b>Valeurs nulles</b>	<table> <thead> <tr> <th></th> <th>Type</th> <th>Val_notnull</th> <th>Val_null</th> <th>%_null</th> </tr> </thead> <tbody> <tr> <td><b>nbv</b></td> <td>object</td> <td>1174142</td> <td>2731</td> <td>0.23</td> </tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>nbv</b>	object	1174142	2731	0.23
	Type	Val_notnull	Val_null	%_null							
<b>nbv</b>	object	1174142	2731	0.23							

<b>Outliers</b>	<p><b>outliers_count outliers_unique</b></p> <p><b>nbv</b> 49288 61 [-1, #ERREUR, -1, 10, 10.0, 11, 11.0, 12, 12...</p> <p>Boxplots pour: nbv</p>																																				
<b>Répartition</b>	<p><b>Count % valeurs</b></p> <p><b>nbv</b></p> <table border="1"> <thead> <tr> <th></th><th>Count</th><th>% valeurs</th></tr> </thead> <tbody> <tr> <td><b>-1</b></td><td>561</td><td>0.0</td></tr> <tr> <td><b>#ERREUR</b></td><td>1</td><td>0.0</td></tr> <tr> <td><b>-1</b></td><td>1669</td><td>0.0</td></tr> <tr> <td><b>0</b></td><td>46729</td><td>4.0</td></tr> <tr> <td><b>0.0</b></td><td>67398</td><td>6.0</td></tr> <tr> <td>...</td><td>...</td><td>...</td></tr> <tr> <td><b>9.0</b></td><td>175</td><td>0.0</td></tr> <tr> <td><b>90</b></td><td>7</td><td>0.0</td></tr> <tr> <td><b>91</b></td><td>1</td><td>0.0</td></tr> <tr> <td><b>99</b></td><td>1</td><td>0.0</td></tr> <tr> <td><b>NaN</b></td><td>2731</td><td>0.0</td></tr> </tbody> </table> <p>71 rows × 2 columns</p>		Count	% valeurs	<b>-1</b>	561	0.0	<b>#ERREUR</b>	1	0.0	<b>-1</b>	1669	0.0	<b>0</b>	46729	4.0	<b>0.0</b>	67398	6.0	...	...	...	<b>9.0</b>	175	0.0	<b>90</b>	7	0.0	<b>91</b>	1	0.0	<b>99</b>	1	0.0	<b>NaN</b>	2731	0.0
	Count	% valeurs																																			
<b>-1</b>	561	0.0																																			
<b>#ERREUR</b>	1	0.0																																			
<b>-1</b>	1669	0.0																																			
<b>0</b>	46729	4.0																																			
<b>0.0</b>	67398	6.0																																			
...	...	...																																			
<b>9.0</b>	175	0.0																																			
<b>90</b>	7	0.0																																			
<b>91</b>	1	0.0																																			
<b>99</b>	1	0.0																																			
<b>NaN</b>	2731	0.0																																			
<b>Remarque</b>	<p>Lorsqu'elles ne représentent pas des NaN, certaines valeurs paraissent aberrantes (on dépasse parfois les 50 voies de circulation).</p>																																				

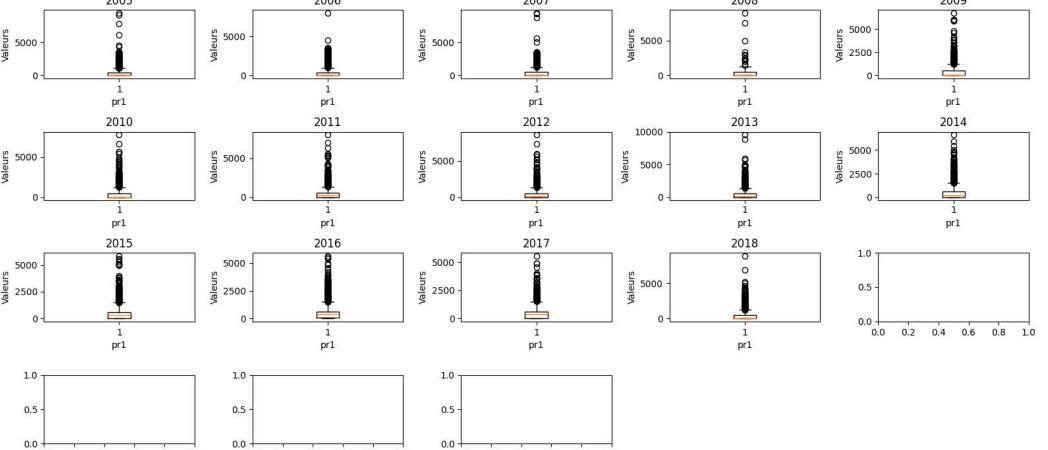
**h. pr**

<b>Description</b>	Numéro du PR de rattachement (numéro de la borne amont). La valeur -
--------------------	--

	1 signifie que le PR n'est pas renseigné.																																	
Type	[2005-2018] : float64 [2019-2022] : object																																	
Etendue des valeurs	<b>count unique top freq</b> <hr/> <b>pr</b> 701389 1413 0.0 150037																																	
Valeurs nulles	<b>Type Val_notnull Val_null %_null</b> <hr/> <b>pr</b> object 701389 475484 40.4																																	
Outliers	<b>outliers_count outliers_unique outliers_list</b> <hr/> <b>pr</b> 349171 1406 [ 0.01, 10, 10.0, 10.2, 10.5, 100, 100.0, 1000...																																	
Répartition	<b>Count % valeurs</b> <hr/> <b>pr</b> <table> <tbody> <tr><td><b>(1)</b></td><td>54073</td><td>5.0</td></tr> <tr><td><b>0</b></td><td>68855</td><td>6.0</td></tr> <tr><td><b>0.0</b></td><td>150037</td><td>13.0</td></tr> <tr><td><b>0.01</b></td><td>1</td><td>0.0</td></tr> <tr><td><b>1</b></td><td>13485</td><td>1.0</td></tr> <tr><td>...</td><td>...</td><td>...</td></tr> <tr><td><b>9900.0</b></td><td>1</td><td>0.0</td></tr> <tr><td><b>992</b></td><td>1</td><td>0.0</td></tr> <tr><td><b>9929.0</b></td><td>1</td><td>0.0</td></tr> <tr><td><b>999</b></td><td>19</td><td>0.0</td></tr> <tr><td><b>NaN</b></td><td>475484</td><td>40.0</td></tr> </tbody> </table> <p>1414 rows × 2 columns</p>	<b>(1)</b>	54073	5.0	<b>0</b>	68855	6.0	<b>0.0</b>	150037	13.0	<b>0.01</b>	1	0.0	<b>1</b>	13485	1.0	...	...	...	<b>9900.0</b>	1	0.0	<b>992</b>	1	0.0	<b>9929.0</b>	1	0.0	<b>999</b>	19	0.0	<b>NaN</b>	475484	40.0
<b>(1)</b>	54073	5.0																																
<b>0</b>	68855	6.0																																
<b>0.0</b>	150037	13.0																																
<b>0.01</b>	1	0.0																																
<b>1</b>	13485	1.0																																
...	...	...																																
<b>9900.0</b>	1	0.0																																
<b>992</b>	1	0.0																																
<b>9929.0</b>	1	0.0																																
<b>999</b>	19	0.0																																
<b>NaN</b>	475484	40.0																																
Remarque	Les valeurs semblent parfois incohérentes, et un grand nombre de NaN prédominent.																																	

### i. pr1

Description	Distance en mètres au PR (par rapport à la borne amont). La valeur -1 signifie que le PR n'est pas renseigné.
Type	[2005-2018] : float64 [2019-2022] : object

Etendue des valeurs	<b>count unique top freq</b> <b>pr1</b> 699570 3708 0.0 198026																																				
Valeurs nulles	<b>Type Val_notnull Val_null %_null</b> <b>pr1</b> object 699570 477303 40.56																																				
Outliers	<b>outliers_count outliers_unique outliers_list</b> <b>pr1</b> 223388 3696 [ 1, 1.0, 10, 10.0, 100, 1000, 1000.0, 1001, 1...  Boxplots pour: pr1 																																				
Répartition	<b>Count % valeurs</b> <b>pr1</b> <table border="1"> <thead> <tr> <th></th> <th>Count</th> <th>% valeurs</th> </tr> </thead> <tbody> <tr> <td><b>(1)</b></td> <td>54819</td> <td>5.0</td> </tr> <tr> <td><b>0</b></td> <td>75650</td> <td>6.0</td> </tr> <tr> <td><b>0.0</b></td> <td>198026</td> <td>17.0</td> </tr> <tr> <td><b>1</b></td> <td>6950</td> <td>1.0</td> </tr> <tr> <td><b>1.0</b></td> <td>9031</td> <td>1.0</td> </tr> <tr> <td><b>...</b></td> <td><b>...</b></td> <td><b>...</b></td> </tr> <tr> <td><b>998</b></td> <td>8</td> <td>0.0</td> </tr> <tr> <td><b>998.0</b></td> <td>41</td> <td>0.0</td> </tr> <tr> <td><b>999</b></td> <td>24</td> <td>0.0</td> </tr> <tr> <td><b>999.0</b></td> <td>258</td> <td>0.0</td> </tr> <tr> <td><b>NaN</b></td> <td>477303</td> <td>41.0</td> </tr> </tbody> </table> <p>3709 rows × 2 columns</p>		Count	% valeurs	<b>(1)</b>	54819	5.0	<b>0</b>	75650	6.0	<b>0.0</b>	198026	17.0	<b>1</b>	6950	1.0	<b>1.0</b>	9031	1.0	<b>...</b>	<b>...</b>	<b>...</b>	<b>998</b>	8	0.0	<b>998.0</b>	41	0.0	<b>999</b>	24	0.0	<b>999.0</b>	258	0.0	<b>NaN</b>	477303	41.0
	Count	% valeurs																																			
<b>(1)</b>	54819	5.0																																			
<b>0</b>	75650	6.0																																			
<b>0.0</b>	198026	17.0																																			
<b>1</b>	6950	1.0																																			
<b>1.0</b>	9031	1.0																																			
<b>...</b>	<b>...</b>	<b>...</b>																																			
<b>998</b>	8	0.0																																			
<b>998.0</b>	41	0.0																																			
<b>999</b>	24	0.0																																			
<b>999.0</b>	258	0.0																																			
<b>NaN</b>	477303	41.0																																			

<b>Remarque</b>	Les valeurs semblent parfois incohérentes, et un grand nombre de NaN prédominent.
-----------------	---

## j. vosp

<b>Description</b>	Signale l'existence d'une voie réservée, indépendamment du fait que l'accident ait lieu ou non sur cette voie.																								
<b>Modalités</b>	<ul style="list-style-type: none"> <li>- -1 : Non renseigné</li> <li>- 0 : Sans objet</li> <li>- 1 : Piste cyclable</li> <li>- 2 : Bande cyclable</li> <li>- 3 : Voie réservée</li> </ul>																								
<b>Type</b>	[2005-2008 ; 2019-2022] : int64 [2009-2018] : float64																								
<b>Etendue des valeurs</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: center;">count</th> <th style="text-align: center;">unique</th> <th style="text-align: center;">top</th> <th style="text-align: center;">freq</th> </tr> </thead> <tbody> <tr> <td><b>vosp</b></td> <td style="text-align: center;">1174112</td> <td style="text-align: center;">5</td> <td style="text-align: center;">0.0</td> <td style="text-align: center;">1090745</td> </tr> </tbody> </table>		count	unique	top	freq	<b>vosp</b>	1174112	5	0.0	1090745														
	count	unique	top	freq																					
<b>vosp</b>	1174112	5	0.0	1090745																					
<b>Valeurs nulles</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: center;">Type</th> <th style="text-align: center;">Val_notnull</th> <th style="text-align: center;">Val_null</th> <th style="text-align: center;">%_null</th> </tr> </thead> <tbody> <tr> <td><b>vosp</b></td> <td style="text-align: center;">float64</td> <td style="text-align: center;">1174112</td> <td style="text-align: center;">2761</td> <td style="text-align: center;">0.23</td> </tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>vosp</b>	float64	1174112	2761	0.23														
	Type	Val_notnull	Val_null	%_null																					
<b>vosp</b>	float64	1174112	2761	0.23																					
<b>Outliers</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: center;">outliers_count</th> <th style="text-align: center;">outliers_unique</th> <th style="text-align: center;">outliers_list</th> </tr> </thead> <tbody> <tr> <td><b>vosp</b></td> <td style="text-align: center;">83367</td> <td style="text-align: center;">4</td> <td style="text-align: center;">[ -1.0, 1.0, 2.0, 3.0 ]</td> </tr> </tbody> </table>		outliers_count	outliers_unique	outliers_list	<b>vosp</b>	83367	4	[ -1.0, 1.0, 2.0, 3.0 ]																
	outliers_count	outliers_unique	outliers_list																						
<b>vosp</b>	83367	4	[ -1.0, 1.0, 2.0, 3.0 ]																						
<b>Répartition</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: center;">Count</th> <th style="text-align: center;">% valeurs</th> </tr> </thead> <tbody> <tr> <td><b>vosp</b></td> <td></td> <td></td> </tr> <tr> <td><b>-1.0</b></td> <td style="text-align: center;">1322</td> <td style="text-align: center;">0.0</td> </tr> <tr> <td><b>0.0</b></td> <td style="text-align: center;">1090745</td> <td style="text-align: center;">93.0</td> </tr> <tr> <td><b>1.0</b></td> <td style="text-align: center;">29795</td> <td style="text-align: center;">3.0</td> </tr> <tr> <td><b>2.0</b></td> <td style="text-align: center;">17853</td> <td style="text-align: center;">2.0</td> </tr> <tr> <td><b>3.0</b></td> <td style="text-align: center;">34397</td> <td style="text-align: center;">3.0</td> </tr> <tr> <td><b>Nan</b></td> <td style="text-align: center;">2761</td> <td style="text-align: center;">0.0</td> </tr> </tbody> </table> <p>The heatmap displays the distribution of 'vosp' values across years (2005-2022) and modalités (0, 1, 2, 3). The y-axis represents the count of values, ranging from 0.0 to 1.0e6. The x-axis represents the modalités. The color scale indicates the year, with 2005 being dark purple and 2022 being yellow. The highest counts are concentrated in modalité 0, with a significant peak in 2005. As the year progresses, the counts generally decrease across all modalités, with a notable presence of values in modalités 1, 2, and 3 in later years.</p>		Count	% valeurs	<b>vosp</b>			<b>-1.0</b>	1322	0.0	<b>0.0</b>	1090745	93.0	<b>1.0</b>	29795	3.0	<b>2.0</b>	17853	2.0	<b>3.0</b>	34397	3.0	<b>Nan</b>	2761	0.0
	Count	% valeurs																							
<b>vosp</b>																									
<b>-1.0</b>	1322	0.0																							
<b>0.0</b>	1090745	93.0																							
<b>1.0</b>	29795	3.0																							
<b>2.0</b>	17853	2.0																							
<b>3.0</b>	34397	3.0																							
<b>Nan</b>	2761	0.0																							

<b>Evolution</b>	
<b>Remarque</b>	Les valeurs NaN peuvent être remplacées par -1 qui signifie « non renseigné ».

## k. prof

<b>Description</b>	Profil en long décrit la déclivité de la route à l'endroit de l'accident.
<b>Modalités</b>	<ul style="list-style-type: none"> <li>- -1 : Non renseigné</li> <li>- 1 : Plat</li> <li>- 2 : Pente</li> <li>- 3 : Sommet de côte</li> <li>- 4 : Bas de côte</li> </ul>
<b>Type</b>	[2005-2008 ; 2019-2022] : int64 [2009-2018] : float64
<b>Etendue des valeurs</b>	<pre>count    unique   top     freq prof  1174924      6  1.0  904058</pre>
<b>Valeurs nulles</b>	<pre>Type  Val_notnull  Val_null %_null prof  float64      1174924    1949    0.17</pre>
<b>Outliers</b>	<pre>outliers_count  outliers_unique      outliers_list prof            270866                  [ -1.0, 0.0, 2.0, 3.0, 4.0 ]</pre>

<b>Répartition</b>  <b>Count % valeurs</b> <b>prof</b> <table border="1"> <thead> <tr> <th></th><th>Count</th><th>% valeurs</th></tr> </thead> <tbody> <tr> <td>-1.0</td><td>38</td><td>0.0</td></tr> <tr> <td>0.0</td><td>65946</td><td>6.0</td></tr> <tr> <td>1.0</td><td>904058</td><td>77.0</td></tr> <tr> <td>2.0</td><td>168117</td><td>14.0</td></tr> <tr> <td>3.0</td><td>20807</td><td>2.0</td></tr> <tr> <td>4.0</td><td>15958</td><td>1.0</td></tr> <tr> <td>NaN</td><td>1949</td><td>0.0</td></tr> </tbody> </table>		Count	% valeurs	-1.0	38	0.0	0.0	65946	6.0	1.0	904058	77.0	2.0	168117	14.0	3.0	20807	2.0	4.0	15958	1.0	NaN	1949	0.0	
	Count	% valeurs																							
-1.0	38	0.0																							
0.0	65946	6.0																							
1.0	904058	77.0																							
2.0	168117	14.0																							
3.0	20807	2.0																							
4.0	15958	1.0																							
NaN	1949	0.0																							
<b>Evolution</b>																									
<b>Remarque</b>	0 n'apparaît pas dans la description. Les valeurs NaN peuvent être remplacées par -1 qui signifie « non renseigné ».																								

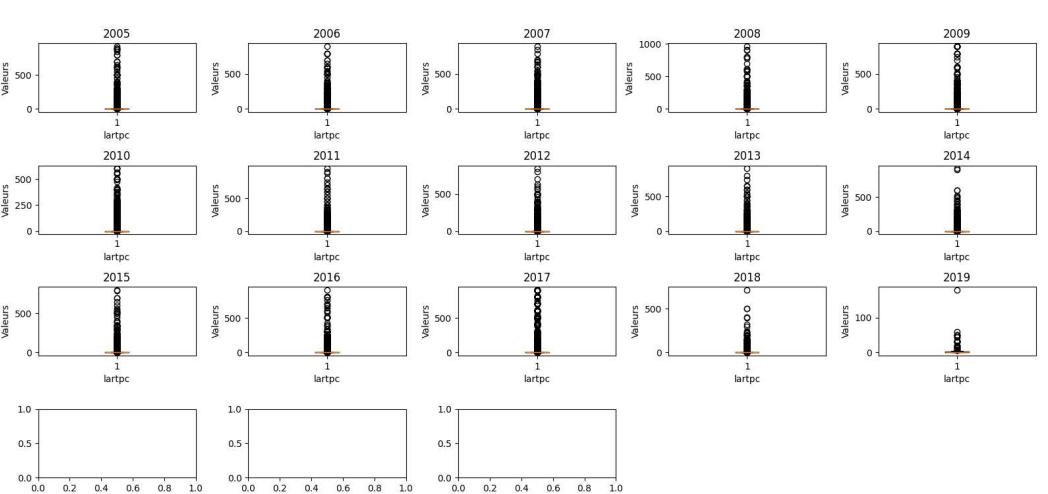
## I. plan

<b>Description</b>	Tracé en plan.										
<b>Modalités</b>	<ul style="list-style-type: none"> <li>- -1 : Non renseigné</li> <li>- 1 : Partie rectiligne</li> <li>- 2 : En courbe à gauche</li> <li>- 3 : En courbe à droite</li> <li>- 4 : En « S »</li> </ul>										
<b>Type</b>	[2005-2008 ; 2019-2022] : int64 [2009-2018] : float64										
<b>Etendue des valeurs</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: center;">count</th> <th style="text-align: center;">unique</th> <th style="text-align: center;">top</th> <th style="text-align: center;">freq</th> </tr> </thead> <tbody> <tr> <td style="text-align: left;"><b>plan</b></td> <td style="text-align: center;">1174592</td> <td style="text-align: center;">6</td> <td style="text-align: center;">1.0</td> <td style="text-align: center;">903356</td> </tr> </tbody> </table>		count	unique	top	freq	<b>plan</b>	1174592	6	1.0	903356
	count	unique	top	freq							
<b>plan</b>	1174592	6	1.0	903356							

Valeurs nulles	Type Val_notnull Val_null %_null																								
	plan float64 1174592 2281 0.19																								
Outliers	outliers_count outliers_unique outliers_list																								
	plan 271236 5 [-1.0, 0.0, 2.0, 3.0, 4.0]																								
Répartition	<p>Count % valeurs</p> <table border="1"> <thead> <tr> <th>plan</th> <th>Count</th> <th>% valeurs</th> </tr> </thead> <tbody> <tr> <td>-1.0</td> <td>34</td> <td>0.0</td> </tr> <tr> <td>0.0</td> <td>66417</td> <td>6.0</td> </tr> <tr> <td>1.0</td> <td>903356</td> <td>77.0</td> </tr> <tr> <td>2.0</td> <td>99412</td> <td>8.0</td> </tr> <tr> <td>3.0</td> <td>90257</td> <td>8.0</td> </tr> <tr> <td>4.0</td> <td>15116</td> <td>1.0</td> </tr> <tr> <td>NaN</td> <td>2281</td> <td>0.0</td> </tr> </tbody> </table> <p>Years: 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022</p>	plan	Count	% valeurs	-1.0	34	0.0	0.0	66417	6.0	1.0	903356	77.0	2.0	99412	8.0	3.0	90257	8.0	4.0	15116	1.0	NaN	2281	0.0
plan	Count	% valeurs																							
-1.0	34	0.0																							
0.0	66417	6.0																							
1.0	903356	77.0																							
2.0	99412	8.0																							
3.0	90257	8.0																							
4.0	15116	1.0																							
NaN	2281	0.0																							
Evolution	<p>Evolution de la distribution plan</p> <p>Modalités: 0, 1, 2, 3, 4, -1</p>																								
Remarque	<p>On observe une modalité 0, non répertoriée qui semble disparaître autour de 2018.</p> <p>Les valeurs NaN peuvent être remplacées par -1 qui signifie « non renseigné ».</p>																								

## m.lartpc

Description	Largeur du terre-plein central (TPC) s'il existe (en m).
Type	[2005-2008] : int64 [2009-2019] : float64

	[2020-2022] : object
Etendue des valeurs	<b>count unique top freq</b> <b>lartpc</b> 902767 711 0.0 479834
Valeurs nulles	<b>Type Val_notnull Val_null %_null</b> <b>lartpc</b> object 902767 274106 23.29
Outliers	<b>outliers_count outliers_unique outliers_list</b> <b>lartpc</b> 108168 707 [ 0,4, 0,8, 1, 1,5, 1,6, 1,0, 1,5, 10, 10,2, 1...  <p style="text-align: center;">Boxplots pour: lartpc</p> 

Répartition	Count	% valeurs
<b>lartpc</b>		
<b>0</b>	279196	24.0
<b>0,4</b>	1	0.0
<b>0,8</b>	1	0.0
<b>0.0</b>	479834	41.0
<b>1</b>	145	0.0
<b>...</b>	<b>...</b>	<b>...</b>
<b>98</b>	21	0.0
<b>98.0</b>	38	0.0
<b>99</b>	12	0.0
<b>99.0</b>	14	0.0
<b>NaN</b>	274106	23.0
712 rows × 2 columns		
Remarque	Les valeurs aberrantes sont très élevées.	

## n. larrouut

Description	Largeur de la chaussée affectée à la circulation des véhicules ne sont pas compris les bandes d'arrêt d'urgence, les TPC et les places de stationnement (en m).
Type	[2009-2019] : float64 [2005-2008] : int64 [2020-2022] : object
Etendue des valeurs	count unique top freq <b>larrouut</b> 1064032 1138 0.0 211137
Valeurs nulles	Type Val_notnull Val_null %_null <b>larrouut</b> object 1064032 112841 9.59
Outliers	outliers_count outliers_unique outliers_list <b>larrouut</b> 417667 1126 [-81, 1, 1, 4, 1.0, 10, 10, 2, 10, 25, 10, 3, 10, ...]

Répartition	Count	% valeurs
<b>larrout</b>		
-1	149480	13.0
-81	1	0.0
0	76818	7.0
0.0	211137	18.0
1	26	0.0
...	...	...
990.0	4	0.0
995	1	0.0
999	2	0.0
999.0	4	0.0
NaN	112841	10.0
1139 rows × 2 columns		
Remarque	Les valeurs existantes semblent très éparpillées.	

## o. surf

Description	Etat de la surface.										
Modalités	<ul style="list-style-type: none"> <li>- -1 : Non renseigné</li> <li>- 1 : Normale</li> <li>- 2 : Mouillée</li> <li>- 3 : Flaque</li> <li>- 4 : Inondée</li> <li>- 5 : Enneigée</li> <li>- 6 : Boue</li> <li>- 7 : Verglacée</li> <li>- 8 : Corps gras – huile</li> <li>- 9 : Autre</li> </ul>										
Type	[2005-2008 ; 2019-2022] : int64 [2009-2018] : float64										
Etendue des valeurs	<table> <thead> <tr> <th></th> <th>count</th> <th>unique</th> <th>top</th> <th>freq</th> </tr> </thead> <tbody> <tr> <td><b>surf</b></td> <td>1174949</td> <td>11</td> <td>1.0</td> <td>921298</td> </tr> </tbody> </table>		count	unique	top	freq	<b>surf</b>	1174949	11	1.0	921298
	count	unique	top	freq							
<b>surf</b>	1174949	11	1.0	921298							
Valeurs nulles	<table> <thead> <tr> <th></th> <th>Type</th> <th>Val_notnull</th> <th>Val_null</th> <th>%_null</th> </tr> </thead> <tbody> <tr> <td><b>surf</b></td> <td>float64</td> <td>1174949</td> <td>1924</td> <td>0.16</td> </tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>surf</b>	float64	1174949	1924	0.16
	Type	Val_notnull	Val_null	%_null							
<b>surf</b>	float64	1174949	1924	0.16							

Outliers	outliers_count outliers_unique			outliers_list																																																				
	surf	253651	10 [-1.0, 0.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0...]																																																					
<b>Répartition</b>																																																								
<b>Count % valeurs</b>																																																								
<b>surf</b>																																																								
<table> <thead> <tr> <th></th> <th>Count</th> <th>% valeurs</th> <th></th> </tr> </thead> <tbody> <tr> <td><b>-1.0</b></td> <td>64</td> <td>0.0</td> <td></td> </tr> <tr> <td><b>0.0</b></td> <td>29139</td> <td>2.0</td> <td></td> </tr> <tr> <td><b>1.0</b></td> <td>921298</td> <td>78.0</td> <td></td> </tr> <tr> <td><b>2.0</b></td> <td>202151</td> <td>17.0</td> <td></td> </tr> <tr> <td><b>3.0</b></td> <td>1671</td> <td>0.0</td> <td></td> </tr> <tr> <td><b>4.0</b></td> <td>580</td> <td>0.0</td> <td></td> </tr> <tr> <td><b>5.0</b></td> <td>3285</td> <td>0.0</td> <td></td> </tr> <tr> <td><b>6.0</b></td> <td>701</td> <td>0.0</td> <td></td> </tr> <tr> <td><b>7.0</b></td> <td>6948</td> <td>1.0</td> <td></td> </tr> <tr> <td><b>8.0</b></td> <td>2735</td> <td>0.0</td> <td></td> </tr> <tr> <td><b>9.0</b></td> <td>6377</td> <td>1.0</td> <td></td> </tr> <tr> <td><b>Nan</b></td> <td>1924</td> <td>0.0</td> <td></td> </tr> </tbody> </table>					Count	% valeurs		<b>-1.0</b>	64	0.0		<b>0.0</b>	29139	2.0		<b>1.0</b>	921298	78.0		<b>2.0</b>	202151	17.0		<b>3.0</b>	1671	0.0		<b>4.0</b>	580	0.0		<b>5.0</b>	3285	0.0		<b>6.0</b>	701	0.0		<b>7.0</b>	6948	1.0		<b>8.0</b>	2735	0.0		<b>9.0</b>	6377	1.0		<b>Nan</b>	1924	0.0		
	Count	% valeurs																																																						
<b>-1.0</b>	64	0.0																																																						
<b>0.0</b>	29139	2.0																																																						
<b>1.0</b>	921298	78.0																																																						
<b>2.0</b>	202151	17.0																																																						
<b>3.0</b>	1671	0.0																																																						
<b>4.0</b>	580	0.0																																																						
<b>5.0</b>	3285	0.0																																																						
<b>6.0</b>	701	0.0																																																						
<b>7.0</b>	6948	1.0																																																						
<b>8.0</b>	2735	0.0																																																						
<b>9.0</b>	6377	1.0																																																						
<b>Nan</b>	1924	0.0																																																						
<b>Evolution</b>																																																								
Remarque	Les valeurs Nan peuvent être remplacées par -1 qui signifie « non renseigné ».																																																							

## p. infra

Description	Aménagement - Infrastructure.
Modalités	- -1 : Non renseigné

	<ul style="list-style-type: none"> <li>- 0 : Aucun</li> <li>- 1 : Souterrain - tunnel</li> <li>- 2 : Pont - autopont</li> <li>- 3 : Bretelle d'échangeur ou de raccordement</li> <li>- 4 : Voie ferrée</li> <li>- 5 : Carrefour aménagé</li> <li>- 6 : Zone piétonne</li> <li>- 7 : Zone de péage</li> <li>- 8 : Chantier</li> <li>- 9 : Autres</li> </ul>																																										
Type	[2005-2008 ; 2019-2002] : int64 [2009-2018] : float64																																										
Etendue des valeurs	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">count</th><th style="text-align: center;">unique</th><th style="text-align: center;">top</th><th style="text-align: center;">freq</th></tr> </thead> <tbody> <tr> <td><b>infra</b></td><td style="text-align: center;">1171465</td><td style="text-align: center;">11</td><td style="text-align: center;">0.0</td><td style="text-align: center;">1032089</td></tr> </tbody> </table>		count	unique	top	freq	<b>infra</b>	1171465	11	0.0	1032089																																
	count	unique	top	freq																																							
<b>infra</b>	1171465	11	0.0	1032089																																							
Valeurs nulles	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">Type</th><th style="text-align: center;">Val_notnull</th><th style="text-align: center;">Val_null</th><th style="text-align: center;">%_null</th></tr> </thead> <tbody> <tr> <td><b>infra</b></td><td style="text-align: center;">float64</td><td style="text-align: center;">1171465</td><td style="text-align: center;">5408</td><td style="text-align: center;">0.46</td></tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>infra</b>	float64	1171465	5408	0.46																																
	Type	Val_notnull	Val_null	%_null																																							
<b>infra</b>	float64	1171465	5408	0.46																																							
Outliers	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">outliers_count</th><th style="text-align: center;">outliers_unique</th><th style="text-align: center;">outliers_list</th></tr> </thead> <tbody> <tr> <td><b>infra</b></td><td style="text-align: center;">139376</td><td style="text-align: center;">10</td><td style="text-align: center;">[-1.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0...]</td></tr> </tbody> </table>		outliers_count	outliers_unique	outliers_list	<b>infra</b>	139376	10	[-1.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0...]																																		
	outliers_count	outliers_unique	outliers_list																																								
<b>infra</b>	139376	10	[-1.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0...]																																								
<b>Répartition</b>  <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">Count</th><th style="text-align: center;">% valeurs</th></tr> </thead> <tbody> <tr> <td><b>infra</b></td><td></td><td></td></tr> <tr> <td><b>-1.0</b></td><td style="text-align: center;">2236</td><td style="text-align: center;">0.0</td></tr> <tr> <td><b>0.0</b></td><td style="text-align: center;">1032089</td><td style="text-align: center;">88.0</td></tr> <tr> <td><b>1.0</b></td><td style="text-align: center;">10754</td><td style="text-align: center;">1.0</td></tr> <tr> <td><b>2.0</b></td><td style="text-align: center;">17283</td><td style="text-align: center;">1.0</td></tr> <tr> <td><b>3.0</b></td><td style="text-align: center;">17768</td><td style="text-align: center;">2.0</td></tr> <tr> <td><b>4.0</b></td><td style="text-align: center;">4113</td><td style="text-align: center;">0.0</td></tr> <tr> <td><b>5.0</b></td><td style="text-align: center;">68524</td><td style="text-align: center;">6.0</td></tr> <tr> <td><b>6.0</b></td><td style="text-align: center;">8329</td><td style="text-align: center;">1.0</td></tr> <tr> <td><b>7.0</b></td><td style="text-align: center;">699</td><td style="text-align: center;">0.0</td></tr> <tr> <td><b>8.0</b></td><td style="text-align: center;">1633</td><td style="text-align: center;">0.0</td></tr> <tr> <td><b>9.0</b></td><td style="text-align: center;">8037</td><td style="text-align: center;">1.0</td></tr> <tr> <td><b>Nan</b></td><td style="text-align: center;">5408</td><td style="text-align: center;">0.0</td></tr> </tbody> </table>		Count	% valeurs	<b>infra</b>			<b>-1.0</b>	2236	0.0	<b>0.0</b>	1032089	88.0	<b>1.0</b>	10754	1.0	<b>2.0</b>	17283	1.0	<b>3.0</b>	17768	2.0	<b>4.0</b>	4113	0.0	<b>5.0</b>	68524	6.0	<b>6.0</b>	8329	1.0	<b>7.0</b>	699	0.0	<b>8.0</b>	1633	0.0	<b>9.0</b>	8037	1.0	<b>Nan</b>	5408	0.0	<p style="text-align: center;">infra</p> <p style="text-align: right;">Years 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022</p>
	Count	% valeurs																																									
<b>infra</b>																																											
<b>-1.0</b>	2236	0.0																																									
<b>0.0</b>	1032089	88.0																																									
<b>1.0</b>	10754	1.0																																									
<b>2.0</b>	17283	1.0																																									
<b>3.0</b>	17768	2.0																																									
<b>4.0</b>	4113	0.0																																									
<b>5.0</b>	68524	6.0																																									
<b>6.0</b>	8329	1.0																																									
<b>7.0</b>	699	0.0																																									
<b>8.0</b>	1633	0.0																																									
<b>9.0</b>	8037	1.0																																									
<b>Nan</b>	5408	0.0																																									

<b>Evolution</b>	
<b>Remarque</b>	Les valeurs NaN peuvent être remplacées par -1 qui signifie « non renseigné ».

### q. situ

<b>Description</b>	Situation de l'accident.										
<b>Modalités</b>	<ul style="list-style-type: none"> <li>- -1 : Non renseigné</li> <li>- 0 : Aucun</li> <li>- 1 : Sur chaussée</li> <li>- 2 : Sur bande d'arrêt d'urgence</li> <li>- 3 : Sur accotement</li> <li>- 4 : Sur trottoir</li> <li>- 5 : Sur piste cyclable</li> <li>- 6 : Sur autre voie spéciale</li> <li>- 8 : Autres</li> </ul>										
<b>Type</b>	[2009-2018] : float64 [2005-2008] : int64 [2019-2022] : object										
<b>Etendue des valeurs</b>	<table> <thead> <tr> <th></th> <th>count</th> <th>unique</th> <th>top</th> <th>freq</th> </tr> </thead> <tbody> <tr> <td><b>situ</b></td> <td>1171903</td> <td>9</td> <td>1.0</td> <td>983318</td> </tr> </tbody> </table>		count	unique	top	freq	<b>situ</b>	1171903	9	1.0	983318
	count	unique	top	freq							
<b>situ</b>	1171903	9	1.0	983318							
<b>Valeurs nulles</b>	<table> <thead> <tr> <th></th> <th>Type</th> <th>Val_notnull</th> <th>Val_null</th> <th>%_null</th> </tr> </thead> <tbody> <tr> <td><b>situ</b></td> <td>float64</td> <td>1171903</td> <td>4970</td> <td>0.42</td> </tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>situ</b>	float64	1171903	4970	0.42
	Type	Val_notnull	Val_null	%_null							
<b>situ</b>	float64	1171903	4970	0.42							
<b>Outliers</b>	<table> <thead> <tr> <th></th> <th>outliers_count</th> <th>outliers_unique</th> <th>outliers_list</th> </tr> </thead> <tbody> <tr> <td><b>situ</b></td> <td>188585</td> <td>8</td> <td>[ -1.0, 0.0, 2.0, 3.0, 4.0, 5.0, 6.0, 8.0 ]</td> </tr> </tbody> </table>		outliers_count	outliers_unique	outliers_list	<b>situ</b>	188585	8	[ -1.0, 0.0, 2.0, 3.0, 4.0, 5.0, 6.0, 8.0 ]		
	outliers_count	outliers_unique	outliers_list								
<b>situ</b>	188585	8	[ -1.0, 0.0, 2.0, 3.0, 4.0, 5.0, 6.0, 8.0 ]								

<h3>Répartition</h3> <table border="1"> <thead> <tr> <th></th> <th>Count</th> <th>% valeurs</th> </tr> </thead> <tbody> <tr> <td><b>situ</b></td> <td></td> <td></td> </tr> <tr> <td><b>-1.0</b></td> <td>142</td> <td>0.0</td> </tr> <tr> <td><b>0.0</b></td> <td>47458</td> <td>4.0</td> </tr> <tr> <td><b>1.0</b></td> <td>983318</td> <td>84.0</td> </tr> <tr> <td><b>2.0</b></td> <td>8456</td> <td>1.0</td> </tr> <tr> <td><b>3.0</b></td> <td>92269</td> <td>8.0</td> </tr> <tr> <td><b>4.0</b></td> <td>20686</td> <td>2.0</td> </tr> <tr> <td><b>5.0</b></td> <td>10424</td> <td>1.0</td> </tr> <tr> <td><b>6.0</b></td> <td>2580</td> <td>0.0</td> </tr> <tr> <td><b>8.0</b></td> <td>6570</td> <td>1.0</td> </tr> <tr> <td><b>NaN</b></td> <td>4970</td> <td>0.0</td> </tr> </tbody> </table>		Count	% valeurs	<b>situ</b>			<b>-1.0</b>	142	0.0	<b>0.0</b>	47458	4.0	<b>1.0</b>	983318	84.0	<b>2.0</b>	8456	1.0	<b>3.0</b>	92269	8.0	<b>4.0</b>	20686	2.0	<b>5.0</b>	10424	1.0	<b>6.0</b>	2580	0.0	<b>8.0</b>	6570	1.0	<b>NaN</b>	4970	0.0	<p>Le histogramme montre la répartition des modalités pour l'année 2022. L'axe des abscisses est étiqueté 'Modalités' et va de 0 à 71. L'axe des ordonnées est étiqueté 'Count' et va de 0.0 à 1.0e6. La couleur des barres indique l'année, avec un gradient qui passe par le bleu, le vert, le jaune et le rouge. La modalité 1 (jaune) est la plus importante, suivie par la modalité 3 (vert).</p>
	Count	% valeurs																																			
<b>situ</b>																																					
<b>-1.0</b>	142	0.0																																			
<b>0.0</b>	47458	4.0																																			
<b>1.0</b>	983318	84.0																																			
<b>2.0</b>	8456	1.0																																			
<b>3.0</b>	92269	8.0																																			
<b>4.0</b>	20686	2.0																																			
<b>5.0</b>	10424	1.0																																			
<b>6.0</b>	2580	0.0																																			
<b>8.0</b>	6570	1.0																																			
<b>NaN</b>	4970	0.0																																			
<h3>Evolution</h3>	<p>Le graphique montre l'évolution de la distribution situ entre 2005 et 2022. L'axe des abscisses est 'Années' et l'axe des ordonnées est 'Proportion (%)'. Il y a huit séries colorées correspondant aux modalités 0 à 7 et -1. La modalité 1 (orange) domine, suivi par la modalité 3 (rouge). Les autres modalités sont très peu représentées.</p>																																				
<h3>Remarque</h3>	<p>On observe de brusques variations des modalités sur accotement et sur chaussée aux alentours de 2019. Les valeurs NaN peuvent être remplacées par -1 qui signifie « non renseigné ».</p>																																				

## r. env1

<b>Description</b>	Point école : proximité d'une école
<b>Modalités</b>	<ul style="list-style-type: none"> <li>- 0</li> <li>- 3</li> <li>- 99</li> </ul>
<b>Type</b>	[2005-2008] : int64 [2009-2018] : float64

<b>Etendue des valeurs</b>	<b>count unique top freq</b>														
	<b>env1</b> 953029 3 0.0 541532														
<b>Valeurs nulles</b>	<b>Type Val_notnull Val_null %_null</b>														
	<b>env1</b> float64 953029 223844 19.02														
<b>Outliers</b>	<b>outliers_count outliers_unique outliers_list</b>														
	<b>env1</b> 0 0 []														
<b>Répartition</b>															
	<b>Count % valeurs</b> <b>env1</b> <table border="1"> <thead> <tr> <th></th> <th>Count</th> <th>% valeurs</th> </tr> </thead> <tbody> <tr> <td>0.0</td> <td>541532</td> <td>46.0</td> </tr> <tr> <td>3.0</td> <td>44824</td> <td>4.0</td> </tr> <tr> <td>99.0</td> <td>366673</td> <td>31.0</td> </tr> <tr> <td>NaN</td> <td>223844</td> <td>19.0</td> </tr> </tbody> </table>		Count	% valeurs	0.0	541532	46.0	3.0	44824	4.0	99.0	366673	31.0	NaN	223844
	Count	% valeurs													
0.0	541532	46.0													
3.0	44824	4.0													
99.0	366673	31.0													
NaN	223844	19.0													
<b>Evolution</b>															
<b>Remarque</b>	La variable disparaît à partir de 2018.														

## S. vma

<b>Description</b>	Vitesse maximale autorisée sur le lieu et au moment de l'accident.
<b>Type</b>	[2019-2022] : int64
<b>Etendue des valeurs</b>	<b>count unique top freq</b>
	<b>vma</b> 218404 47 50.0 115319

Valeurs nulles	Type	Val_notnull	Val_null	%_null		
	vma	float64	218404	958469 81.44		
Outliers		outliers_count	outliers_unique	outliers_list		
	vma	7701	21 [-1.0, 0.0, 1.0, 2.0, 3.0, 4.0, 130.0, 140.0,...			
		Boxplots pour: vma				
Répartition	Count	% valeurs				
	vma					
	-1.0	3393 0.29				
	0.0	1 0.0				
	1.0	67 0.01				
	2.0	47 0.0				
	3.0	9 0.0				
	...	...				
	770.0	1 0.0				
	800.0	1 0.0				
	900.0	4 0.0				
	901.0	1 0.0				
	Nan	958469 81.44				
48 rows x 2 columns						
Remarque	La variable apparaît en 2019. Lorsqu'elle n'évoque pas des situations aberrantes, elle correspond souvent à des NaN.					

### 3. Usagers

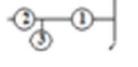
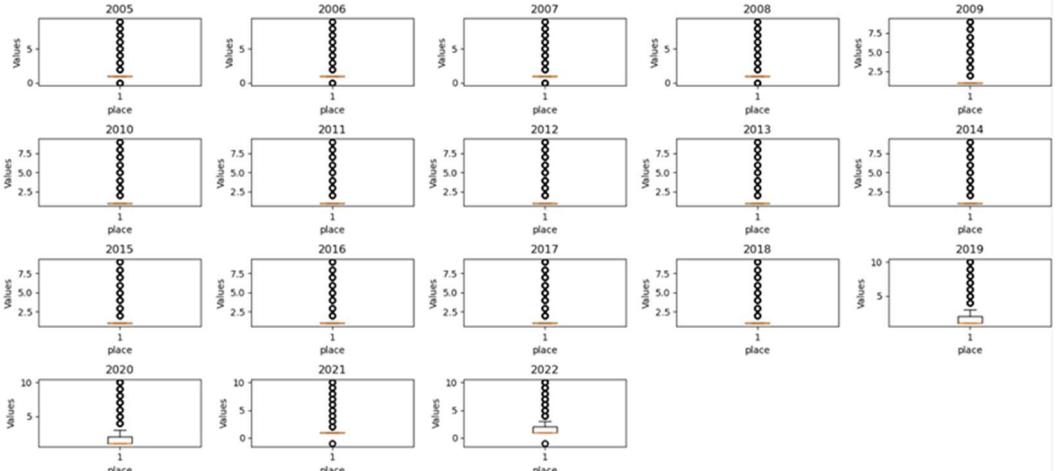
Rows x columns Rows duplicated

**Usagers** (2636377, 17) 2858

#### a. Num\_Acc

Description	Identifiant de l'accident identique à celui du fichier "rubrique CARACTERISTIQUES" repris pour chacun des usagers décrits impliqués dans l'accident.																																																														
Type	int64																																																														
Etendue des valeurs	<table> <thead> <tr> <th></th><th>count</th><th>unique</th><th>top</th><th>freq</th></tr> </thead> <tbody> <tr> <td>Num_Acc</td><td>2636377</td><td>1176873</td><td>200600016834</td><td>86</td></tr> </tbody> </table>						count	unique	top	freq	Num_Acc	2636377	1176873	200600016834	86																																																
	count	unique	top	freq																																																											
Num_Acc	2636377	1176873	200600016834	86																																																											
Valeurs nulles	<table> <thead> <tr> <th></th><th>Type</th><th>Val_notnull</th><th>Val_null</th><th>%_null</th></tr> </thead> <tbody> <tr> <td>Num_Acc</td><td>int64</td><td>2636377</td><td>0</td><td>0.0</td></tr> </tbody> </table>						Type	Val_notnull	Val_null	%_null	Num_Acc	int64	2636377	0	0.0																																																
	Type	Val_notnull	Val_null	%_null																																																											
Num_Acc	int64	2636377	0	0.0																																																											
Outliers	<table> <thead> <tr> <th></th><th>outliers_count</th><th>outliers_unique</th><th>outliers_list</th><th></th></tr> </thead> <tbody> <tr> <td>Num_Acc</td><td>0</td><td>0</td><td>[ ]</td><td></td></tr> </tbody> </table>						outliers_count	outliers_unique	outliers_list		Num_Acc	0	0	[ ]																																																	
	outliers_count	outliers_unique	outliers_list																																																												
Num_Acc	0	0	[ ]																																																												
Répartition	<table> <thead> <tr> <th></th><th>Count</th><th>% valeurs</th><th></th></tr> <tr> <th>Num_Acc</th><th></th><th></th><th></th></tr> </thead> <tbody> <tr> <td>200500000001</td><td>6</td><td>0.0</td><td></td></tr> <tr> <td>200500000002</td><td>2</td><td>0.0</td><td></td></tr> <tr> <td>200500000003</td><td>2</td><td>0.0</td><td></td></tr> <tr> <td>200500000004</td><td>4</td><td>0.0</td><td></td></tr> <tr> <td>200500000005</td><td>2</td><td>0.0</td><td></td></tr> <tr> <td>...</td><td>...</td><td>...</td><td></td></tr> <tr> <td>202200055298</td><td>2</td><td>0.0</td><td></td></tr> <tr> <td>202200055299</td><td>1</td><td>0.0</td><td></td></tr> <tr> <td>202200055300</td><td>1</td><td>0.0</td><td></td></tr> <tr> <td>202200055301</td><td>3</td><td>0.0</td><td></td></tr> <tr> <td>202200055302</td><td>2</td><td>0.0</td><td></td></tr> <tr> <td colspan="4">1176873 rows × 2 columns</td><td></td><td></td></tr> </tbody> </table>						Count	% valeurs		Num_Acc				200500000001	6	0.0		200500000002	2	0.0		200500000003	2	0.0		200500000004	4	0.0		200500000005	2	0.0		...	...	...		202200055298	2	0.0		202200055299	1	0.0		202200055300	1	0.0		202200055301	3	0.0		202200055302	2	0.0		1176873 rows × 2 columns					
	Count	% valeurs																																																													
Num_Acc																																																															
200500000001	6	0.0																																																													
200500000002	2	0.0																																																													
200500000003	2	0.0																																																													
200500000004	4	0.0																																																													
200500000005	2	0.0																																																													
...	...	...																																																													
202200055298	2	0.0																																																													
202200055299	1	0.0																																																													
202200055300	1	0.0																																																													
202200055301	3	0.0																																																													
202200055302	2	0.0																																																													
1176873 rows × 2 columns																																																															

## b. place

Description	Permet de situer la place occupée dans le véhicule par l'usager au moment de l'accident.																																			
	Transport en commun																																			
	Voiture																																			
	Moto / Side-car																																			
	 <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>4</td><td>7</td><td>7</td><td>7</td><td></td><td></td><td></td></tr> <tr><td>5</td><td>8</td><td>8</td><td>8</td><td></td><td>7</td><td>7</td></tr> <tr><td>5</td><td>8</td><td>8</td><td>8</td><td></td><td>8</td><td>8</td></tr> <tr><td>5</td><td>8</td><td>8</td><td>8</td><td></td><td>8</td><td>8</td></tr> <tr><td>3</td><td>9</td><td>9</td><td>9</td><td></td><td>9</td><td>9</td></tr> </table>	4	7	7	7				5	8	8	8		7	7	5	8	8	8		8	8	5	8	8	8		8	8	3	9	9	9		9	9
4	7	7	7																																	
5	8	8	8		7	7																														
5	8	8	8		8	8																														
5	8	8	8		8	8																														
3	9	9	9		9	9																														
Type	[2005-2008 ; 2019-2022] : int64 [2009-2018] : float64																																			
Etendue des valeurs	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: left;">count</th> <th style="text-align: left;">unique</th> <th style="text-align: left;">top</th> <th style="text-align: left;">freq</th> </tr> </thead> <tbody> <tr> <td>place</td> <td>2513249</td> <td>12</td> <td>1.0</td> <td>1962529</td> </tr> </tbody> </table>		count	unique	top	freq	place	2513249	12	1.0	1962529																									
	count	unique	top	freq																																
place	2513249	12	1.0	1962529																																
Valeurs nulles	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: left;">Type</th> <th style="text-align: left;">Val_notnull</th> <th style="text-align: left;">Val_null</th> <th style="text-align: left;">%_null</th> </tr> </thead> <tbody> <tr> <td>place</td> <td>float64</td> <td>2513249</td> <td>123128</td> <td>4.67</td> </tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	place	float64	2513249	123128	4.67																									
	Type	Val_notnull	Val_null	%_null																																
place	float64	2513249	123128	4.67																																
Outliers	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: left;">outliers_count</th> <th style="text-align: left;">outliers_unique</th> <th style="text-align: right;">outliers_list</th> </tr> </thead> <tbody> <tr> <td>place</td> <td>550720</td> <td>11</td> <td style="text-align: right;">[-1.0, 0.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0...]</td> </tr> </tbody> </table> <p style="text-align: center;">Boxplots for column: place</p> 		outliers_count	outliers_unique	outliers_list	place	550720	11	[-1.0, 0.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0...]																											
	outliers_count	outliers_unique	outliers_list																																	
place	550720	11	[-1.0, 0.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0...]																																	

<b>Répartition</b>	<table border="1"> <thead> <tr> <th>Modalité</th><th>Count</th><th>% valeurs</th></tr> </thead> <tbody> <tr><td>-1.0</td><td>25</td><td>0.0</td></tr> <tr><td>0.0</td><td>60766</td><td>2.0</td></tr> <tr><td>1.0</td><td>1962529</td><td>78.0</td></tr> <tr><td>2.0</td><td>281895</td><td>11.0</td></tr> <tr><td>3.0</td><td>60272</td><td>2.0</td></tr> <tr><td>4.0</td><td>52156</td><td>2.0</td></tr> <tr><td>5.0</td><td>27703</td><td>1.0</td></tr> <tr><td>6.0</td><td>2647</td><td>0.0</td></tr> <tr><td>7.0</td><td>9362</td><td>0.0</td></tr> <tr><td>8.0</td><td>8118</td><td>0.0</td></tr> <tr><td>9.0</td><td>9184</td><td>0.0</td></tr> <tr><td>10.0</td><td>38592</td><td>2.0</td></tr> </tbody> </table>	Modalité	Count	% valeurs	-1.0	25	0.0	0.0	60766	2.0	1.0	1962529	78.0	2.0	281895	11.0	3.0	60272	2.0	4.0	52156	2.0	5.0	27703	1.0	6.0	2647	0.0	7.0	9362	0.0	8.0	8118	0.0	9.0	9184	0.0	10.0	38592	2.0
Modalité	Count	% valeurs																																						
-1.0	25	0.0																																						
0.0	60766	2.0																																						
1.0	1962529	78.0																																						
2.0	281895	11.0																																						
3.0	60272	2.0																																						
4.0	52156	2.0																																						
5.0	27703	1.0																																						
6.0	2647	0.0																																						
7.0	9362	0.0																																						
8.0	8118	0.0																																						
9.0	9184	0.0																																						
10.0	38592	2.0																																						
<b>Evolution</b>																																								
<b>Remarque</b>	Certaines modalités (-1 et 0) ne sont pas répertoriées dans la description.																																							

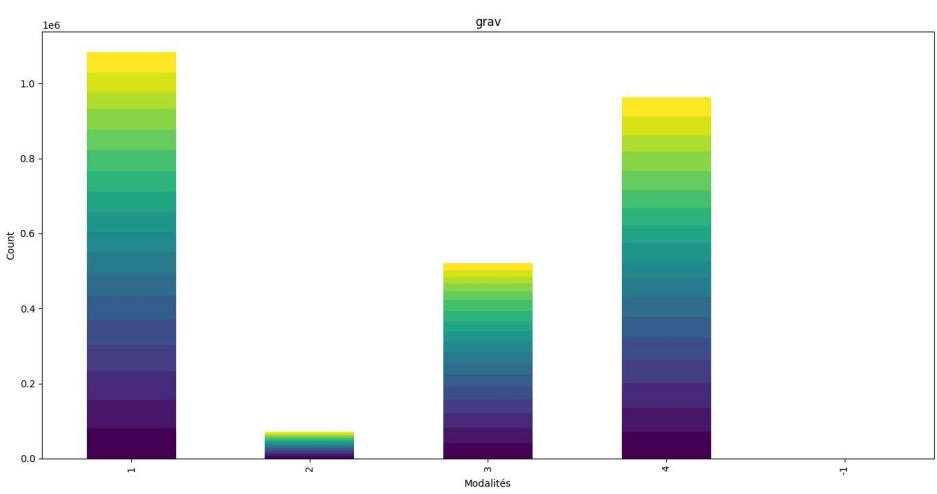
### c. catu

<b>Description</b>	Catégorie d'usager.										
<b>Modalités</b>	<ul style="list-style-type: none"> <li>- 1 : Conducteur</li> <li>- 2 : Passager</li> <li>- 3 : Piéton</li> </ul>										
<b>Type</b>	int64										
<b>Etendue des valeurs</b>	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th></th> <th>count</th> <th>unique</th> <th>top</th> <th>freq</th> </tr> </thead> <tbody> <tr> <td><b>catu</b></td> <td>2636377</td> <td>4</td> <td>1</td> <td>1961486</td> </tr> </tbody> </table>		count	unique	top	freq	<b>catu</b>	2636377	4	1	1961486
	count	unique	top	freq							
<b>catu</b>	2636377	4	1	1961486							

Valeurs nulles	Type	Val_notnull	Val_null	%_null		
	<b>catu</b>	int64	2636377	0 0.0		
Outliers	outliers_count	outliers_unique	outliers_list			
	<b>catu</b>	3560	1	[ 4.0]		
	Boxplots pour: catu					
Répartition	Count	% valeurs				
Modalité						
<b>1</b>	1961486	74.0				
<b>2</b>	454622	17.0				
<b>3</b>	216709	8.0				
<b>4</b>	3560	0.0				
Evolution	Evolution de la distribution catu					

<b>Remarque</b>	La modalité 4 non répertoriée dans la description, finit par disparaître à partir de 2018.
-----------------	--

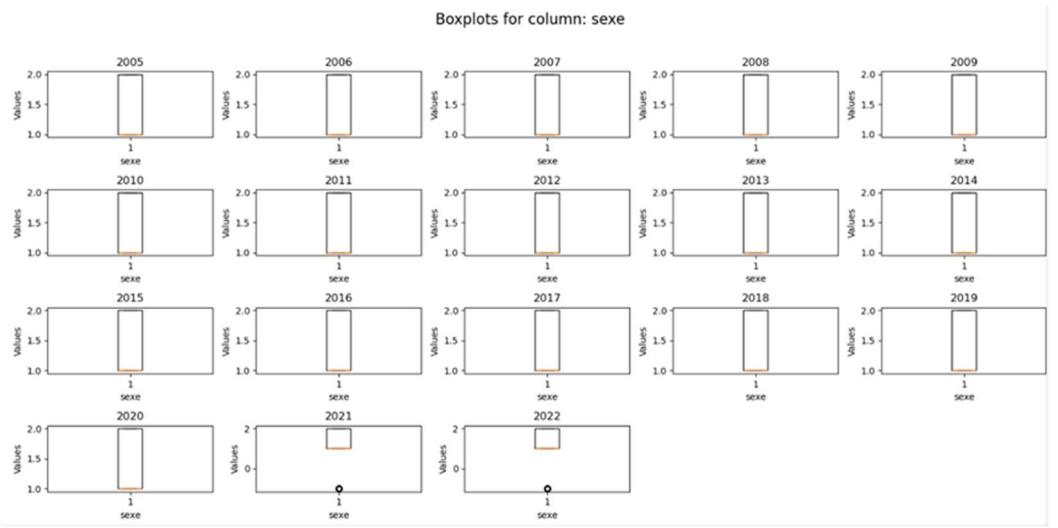
## d. grav

<b>Description</b>	Gravité de blessure de l'usager.																					
<b>Modalités</b>	<ul style="list-style-type: none"> <li>- 1 : Indemne</li> <li>- 2 : Tué</li> <li>- 3 : Blessé hospitalisé</li> <li>- 4 : Blessé léger</li> </ul>																					
<b>Type</b>	int64																					
<b>Valeurs nulles</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th>Type</th> <th>Val_notnull</th> <th>Val_null</th> <th>%_null</th> </tr> </thead> <tbody> <tr> <td>grav</td> <td>int64</td> <td>2636377</td> <td>0</td> <td>0.0</td> </tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	grav	int64	2636377	0	0.0											
	Type	Val_notnull	Val_null	%_null																		
grav	int64	2636377	0	0.0																		
<b>Etendue des valeurs</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th>count</th> <th>unique</th> <th>top</th> <th>freq</th> </tr> </thead> <tbody> <tr> <td>grav</td> <td>2636377</td> <td>5</td> <td>1</td> <td>1082746</td> </tr> </tbody> </table>		count	unique	top	freq	grav	2636377	5	1	1082746											
	count	unique	top	freq																		
grav	2636377	5	1	1082746																		
<b>Outliers</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th>outliers_count</th> <th>outliers_unique</th> <th>outliers_list</th> </tr> </thead> <tbody> <tr> <td>grav</td> <td>0</td> <td>0</td> <td>[]</td> </tr> </tbody> </table>		outliers_count	outliers_unique	outliers_list	grav	0	0	[]													
	outliers_count	outliers_unique	outliers_list																			
grav	0	0	[]																			
<b>Répartition</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th>Count</th> <th>% valeurs</th> </tr> </thead> <tbody> <tr> <td><b>Modalité</b></td> <td></td> <td></td> </tr> <tr> <td>-1</td> <td>301</td> <td>0.0</td> </tr> <tr> <td>1</td> <td>1082746</td> <td>41.0</td> </tr> <tr> <td>2</td> <td>70628</td> <td>3.0</td> </tr> <tr> <td>3</td> <td>520817</td> <td>20.0</td> </tr> <tr> <td>4</td> <td>961885</td> <td>36.0</td> </tr> </tbody> </table> 		Count	% valeurs	<b>Modalité</b>			-1	301	0.0	1	1082746	41.0	2	70628	3.0	3	520817	20.0	4	961885	36.0
	Count	% valeurs																				
<b>Modalité</b>																						
-1	301	0.0																				
1	1082746	41.0																				
2	70628	3.0																				
3	520817	20.0																				
4	961885	36.0																				

<b>Evolution</b>	<p style="text-align: center;">Evolution de la distribution grav</p> <p style="text-align: center;">Années</p>
<b>Remarque</b>	Autour de 2018, un changement de saisie a lieu sur les modalités 4 et 3 (pour la comptabilisation des hospitalisations). La modalité -1 n'est pas répertoriée dans la description.

### e. sexe

<b>Description</b>	Sexe de l'usager										
<b>Modalités</b>	- 1 : Masculin - 2 : Féminin										
<b>Type</b>	int64										
<b>Etendue des valeurs</b>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: center;">count</th> <th style="text-align: center;">unique</th> <th style="text-align: center;">top</th> <th style="text-align: center;">freq</th> </tr> </thead> <tbody> <tr> <td><b>sexé</b></td> <td style="text-align: center;">2636377</td> <td style="text-align: center;">3</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1773190</td> </tr> </tbody> </table>		count	unique	top	freq	<b>sexé</b>	2636377	3	1	1773190
	count	unique	top	freq							
<b>sexé</b>	2636377	3	1	1773190							
<b>Valeurs nulles</b>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: center;">Type</th> <th style="text-align: center;">Val_notnull</th> <th style="text-align: center;">Val_null</th> <th style="text-align: center;">%_null</th> </tr> </thead> <tbody> <tr> <td><b>sexé</b></td> <td style="text-align: center;">int64</td> <td style="text-align: center;">2636377</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0.0</td> </tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>sexé</b>	int64	2636377	0	0.0
	Type	Val_notnull	Val_null	%_null							
<b>sexé</b>	int64	2636377	0	0.0							
<b>Outliers</b>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: center;">outliers_count</th> <th style="text-align: center;">outliers_unique</th> <th style="text-align: center;">outliers_list</th> </tr> </thead> <tbody> <tr> <td><b>sexé</b></td> <td style="text-align: center;">5806</td> <td style="text-align: center;">1</td> <td style="text-align: center;">[ -1.0]</td> </tr> </tbody> </table>		outliers_count	outliers_unique	outliers_list	<b>sexé</b>	5806	1	[ -1.0]		
	outliers_count	outliers_unique	outliers_list								
<b>sexé</b>	5806	1	[ -1.0]								

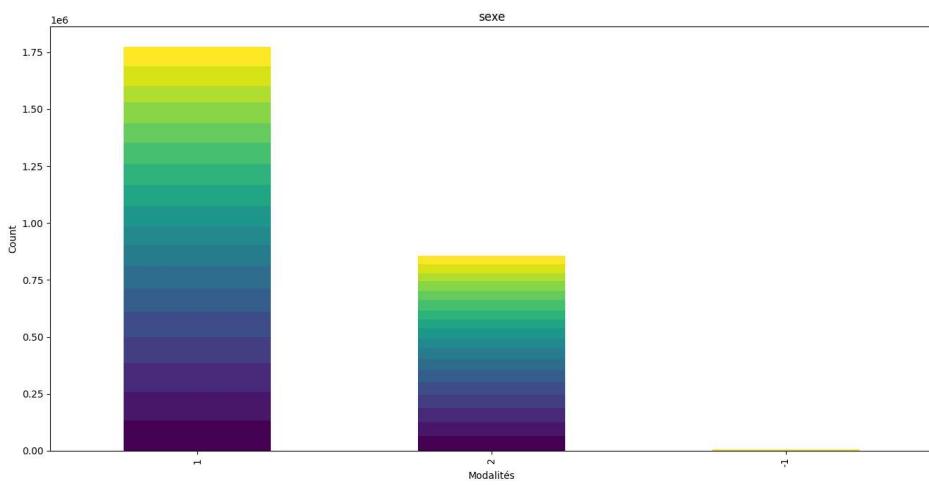


### Répartition

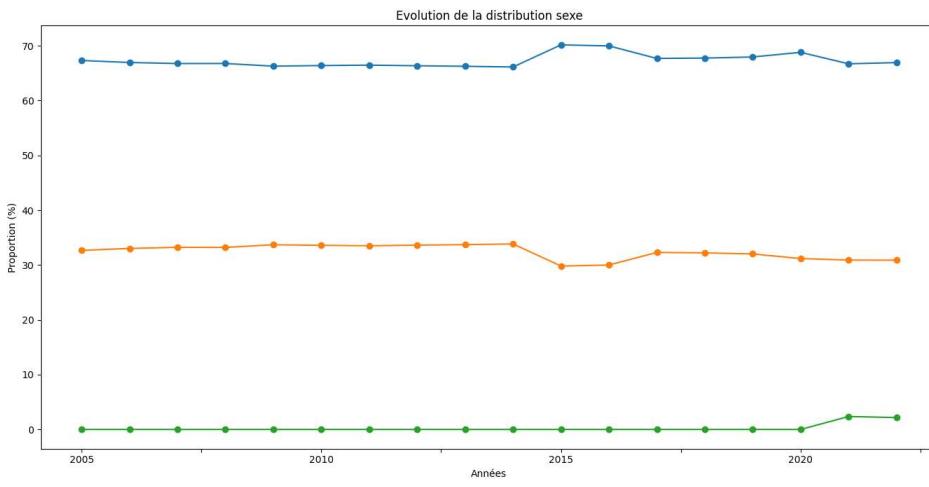
Count % valeurs

#### Modalité

-1	5806	0.0
1	1773190	67.0
2	857381	33.0



### Evolution



### Remarque

La valeur -1 n'est pas répertoriée dans la description et apparaît comme outlier.

## f. trajet

### Description

Motif du déplacement au moment de l'accident.

Modalités	- -1 : Non renseigné - 0 : Non renseigné - 1 : Domicile - travail - 2 : Domicile - école - 3 : Courses - achats - 4 : Utilisation professionnelle - 5 : Promenade - loisirs - 9 : Autre																														
Type	[2005-2008 ; 2019-2022] : int64 [2009-2018] : float64																														
Etendue des valeurs	<table> <thead> <tr> <th></th><th>count</th><th>unique</th><th>top</th><th>freq</th></tr> </thead> <tbody> <tr> <td>trajet</td><td>2635883</td><td>8</td><td>5.0</td><td>978415</td></tr> </tbody> </table>		count	unique	top	freq	trajet	2635883	8	5.0	978415																				
	count	unique	top	freq																											
trajet	2635883	8	5.0	978415																											
Valeurs nulles	<table> <thead> <tr> <th></th><th>Type</th><th>Val_notnull</th><th>Val_null</th><th>%_null</th></tr> </thead> <tbody> <tr> <td>trajet</td><td>float64</td><td>2635883</td><td>494</td><td>0.02</td></tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	trajet	float64	2635883	494	0.02																				
	Type	Val_notnull	Val_null	%_null																											
trajet	float64	2635883	494	0.02																											
Outliers	<table> <thead> <tr> <th></th><th>outliers_count</th><th>outliers_unique</th><th>outliers_list</th></tr> </thead> <tbody> <tr> <td>trajet</td><td>0</td><td>0</td><td>[]</td></tr> </tbody> </table>		outliers_count	outliers_unique	outliers_list	trajet	0	0	[]																						
	outliers_count	outliers_unique	outliers_list																												
trajet	0	0	[]																												
Répartition	<table> <thead> <tr> <th></th><th>Count</th><th>% valeurs</th></tr> </thead> <tbody> <tr> <td>Modalité</td><td></td><td></td></tr> <tr> <td>-1.0</td><td>6900</td><td>0.0</td></tr> <tr> <td>0.0</td><td>734734</td><td>28.0</td></tr> <tr> <td>1.0</td><td>344904</td><td>13.0</td></tr> <tr> <td>2.0</td><td>54763</td><td>2.0</td></tr> <tr> <td>3.0</td><td>71040</td><td>3.0</td></tr> <tr> <td>4.0</td><td>255752</td><td>10.0</td></tr> <tr> <td>5.0</td><td>978415</td><td>37.0</td></tr> <tr> <td>9.0</td><td>189375</td><td>7.0</td></tr> </tbody> </table>		Count	% valeurs	Modalité			-1.0	6900	0.0	0.0	734734	28.0	1.0	344904	13.0	2.0	54763	2.0	3.0	71040	3.0	4.0	255752	10.0	5.0	978415	37.0	9.0	189375	7.0
	Count	% valeurs																													
Modalité																															
-1.0	6900	0.0																													
0.0	734734	28.0																													
1.0	344904	13.0																													
2.0	54763	2.0																													
3.0	71040	3.0																													
4.0	255752	10.0																													
5.0	978415	37.0																													
9.0	189375	7.0																													

<b>Evolution</b>	
<b>Remarque</b>	Les valeurs NaN peuvent être remplacées par -1 qui signifie « non renseigné ».

### g. secu

<b>Description</b>	Enseigne variable sur 2 caractères : - le premier concerne l'existence d'un Équipement de sécurité 1 - Ceinture 2 - Casque 3 - Dispositif enfants 4 - Equipement réfléchissant 9 - Autre - le second concerne l'utilisation de l'Équipement de sécurité 1 - Oui 2 - Non 3 - Non déterminable										
<b>Type</b>	[2005 ;2007-2008] : int64 [2006 ; 2009-2018] : float64										
<b>Etendue des valeurs</b>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: left;">count</th> <th style="text-align: left;">unique</th> <th style="text-align: left;">top</th> <th style="text-align: left;">freq</th> </tr> </thead> <tbody> <tr> <td><b>secu</b></td> <td>2085658</td> <td>24</td> <td>11.0</td> <td>1197467</td> </tr> </tbody> </table>		count	unique	top	freq	<b>secu</b>	2085658	24	11.0	1197467
	count	unique	top	freq							
<b>secu</b>	2085658	24	11.0	1197467							
<b>Valeurs nulles</b>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: left;">Type</th> <th style="text-align: left;">Val_notnull</th> <th style="text-align: left;">Val_null</th> <th style="text-align: left;">%_null</th> </tr> </thead> <tbody> <tr> <td><b>secu</b></td> <td>float64</td> <td>2085658</td> <td>550719</td> <td>20.89</td> </tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>secu</b>	float64	2085658	550719	20.89
	Type	Val_notnull	Val_null	%_null							
<b>secu</b>	float64	2085658	550719	20.89							
<b>Outliers</b>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: left;">outliers_count</th> <th style="text-align: left;">outliers_unique</th> <th style="text-align: right;">outliers_list</th> </tr> </thead> <tbody> <tr> <td><b>secu</b></td> <td>126934</td> <td>8</td> <td style="text-align: right;">[ 40.0, 41.0, 42.0, 43.0, 90.0, 91.0, 92.0, 93.0]</td> </tr> </tbody> </table>		outliers_count	outliers_unique	outliers_list	<b>secu</b>	126934	8	[ 40.0, 41.0, 42.0, 43.0, 90.0, 91.0, 92.0, 93.0]		
	outliers_count	outliers_unique	outliers_list								
<b>secu</b>	126934	8	[ 40.0, 41.0, 42.0, 43.0, 90.0, 91.0, 92.0, 93.0]								

	<p style="text-align: center;">Boxplots for column: secu</p>																																			
<b>Répartition</b>  <b>Count % valeurs</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>secu</th> <th>Count</th> <th>% valeurs</th> </tr> </thead> <tbody> <tr><td>0.0</td><td>68374</td><td>2.59</td></tr> <tr><td>1.0</td><td>3568</td><td>0.14</td></tr> <tr><td>2.0</td><td>2669</td><td>0.1</td></tr> <tr><td>3.0</td><td>7707</td><td>0.29</td></tr> <tr><td>10.0</td><td>5631</td><td>0.21</td></tr> <tr><td>...</td><td>...</td><td>...</td></tr> <tr><td>90.0</td><td>73</td><td>0.0</td></tr> <tr><td>91.0</td><td>7653</td><td>0.29</td></tr> <tr><td>92.0</td><td>7693</td><td>0.29</td></tr> <tr><td>93.0</td><td>105121</td><td>3.99</td></tr> <tr><td>NaN</td><td>550719</td><td>20.89</td></tr> </tbody> </table> <p>25 rows × 2 columns</p>	secu	Count	% valeurs	0.0	68374	2.59	1.0	3568	0.14	2.0	2669	0.1	3.0	7707	0.29	10.0	5631	0.21	...	...	...	90.0	73	0.0	91.0	7653	0.29	92.0	7693	0.29	93.0	105121	3.99	NaN	550719	20.89
secu	Count	% valeurs																																		
0.0	68374	2.59																																		
1.0	3568	0.14																																		
2.0	2669	0.1																																		
3.0	7707	0.29																																		
10.0	5631	0.21																																		
...	...	...																																		
90.0	73	0.0																																		
91.0	7653	0.29																																		
92.0	7693	0.29																																		
93.0	105121	3.99																																		
NaN	550719	20.89																																		
<b>Remarque</b>	La variable disparaît à partir de 2019 au profit de secu1/2/3. Forte proportion de valeurs NaN.																																			

## h. locp

<b>Description</b>	Localisation du piéton.
<b>Modalités</b>	<ul style="list-style-type: none"> <li>- -1 : Non renseigné</li> <li>- 0 : Sans objet</li> <li>- 1 : Sur chaussée : A + 50 m du passage piéton</li> </ul>

	<ul style="list-style-type: none"> <li>- 2 : Sur chaussée : A - 50 m du passage piéton</li> <li>- 3 : Sur passage piéton : Sans signalisation lumineuse</li> <li>- 4 : Sur passage piéton : Avec signalisation lumineuse</li> <li>- 5 : Sur trottoir</li> <li>- 6 : Sur accotement</li> <li>- 7 : Sur refuge ou BAU</li> <li>- 8 : Sur contre allée</li> <li>- 9 : Inconnue</li> </ul>										
Type	[2005-2008 ; 2019-2022] : int64 [2009-2018] : float64										
Etendue des valeurs	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">count</th><th style="text-align: center;">unique</th><th style="text-align: center;">top</th><th style="text-align: center;">freq</th></tr> </thead> <tbody> <tr> <td style="color: #0070C0;"><b>locp</b></td><td style="color: #0070C0;">2580016</td><td style="color: #0070C0;">11</td><td style="color: #0070C0;">0.0</td><td style="color: #0070C0;">2162665</td></tr> </tbody> </table>		count	unique	top	freq	<b>locp</b>	2580016	11	0.0	2162665
	count	unique	top	freq							
<b>locp</b>	2580016	11	0.0	2162665							
Valeurs nulles	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">Type</th><th style="text-align: center;">Val_notnull</th><th style="text-align: center;">Val_null</th><th style="text-align: center;">%_null</th></tr> </thead> <tbody> <tr> <td style="color: #0070C0;"><b>locp</b></td><td style="color: #0070C0;">float64</td><td style="color: #0070C0;">2580016</td><td style="color: #0070C0;">56361</td><td style="color: #0070C0;">2.14</td></tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>locp</b>	float64	2580016	56361	2.14
	Type	Val_notnull	Val_null	%_null							
<b>locp</b>	float64	2580016	56361	2.14							
Outliers	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">outliers_count</th><th style="text-align: center;">outliers_unique</th><th style="text-align: right; vertical-align: bottom;">outliers_list</th></tr> </thead> <tbody> <tr> <td style="color: #0070C0;"><b>locp</b></td><td style="color: #0070C0;">417351</td><td style="color: #0070C0;">10</td><td style="text-align: right; vertical-align: bottom;">[-1.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0...]</td></tr> </tbody> </table> <p style="text-align: center;">Boxplots for column: locp</p>		outliers_count	outliers_unique	outliers_list	<b>locp</b>	417351	10	[-1.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0...]		
	outliers_count	outliers_unique	outliers_list								
<b>locp</b>	417351	10	[-1.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0...]								

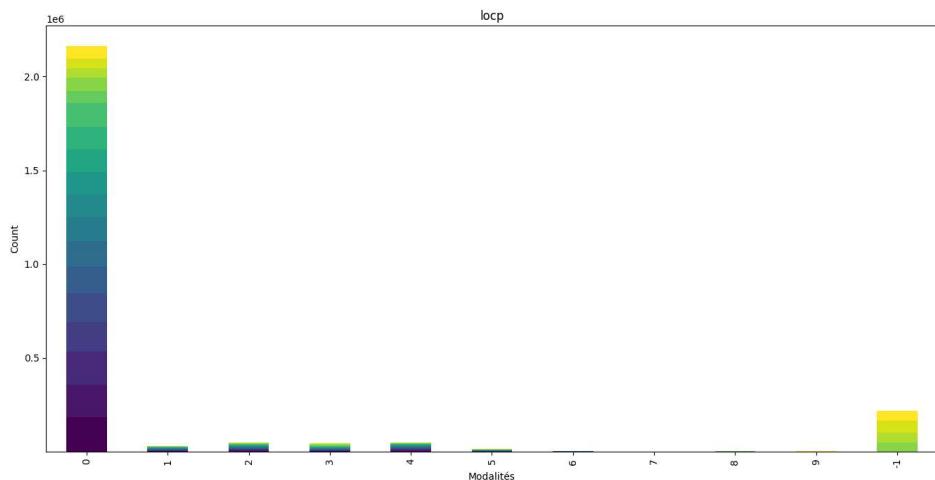
## Répartition

Count % valeurs

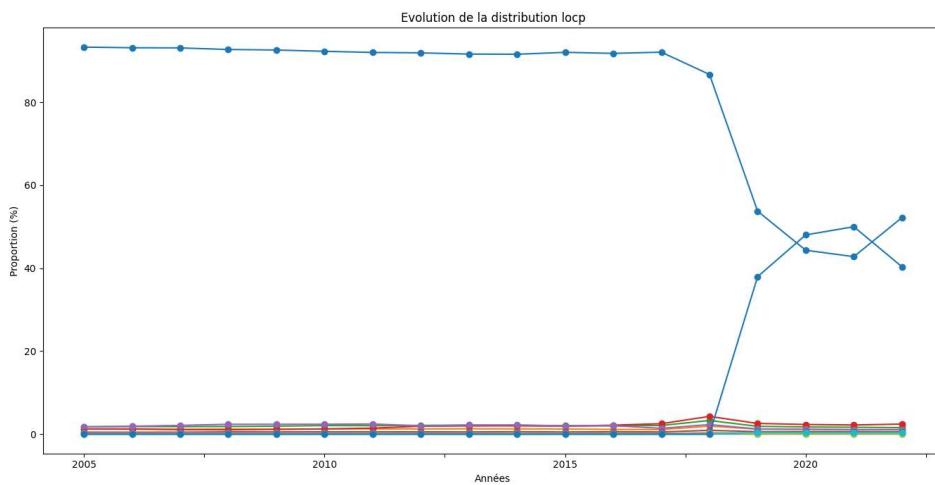
locp

	Count	% valeurs
-1.0	216753	8.22
0.0	2162665	82.03
1.0	31299	1.19
2.0	50467	1.91
3.0	46270	1.76
...	...	...
6.0	4703	0.18
7.0	247	0.01
8.0	2367	0.09
9.0	1844	0.07
NaN	56361	2.14

12 rows × 2 columns



## Evolution



## Remarque

Attention aux valeurs -1 et 0 pour lesquelles il pourrait y avoir un amalgame.  
Les valeurs NaN peuvent être remplacées par -1 qui signifie « non renseigné ».

## i. actp

### Description

Action du piéton.

### Modalités

- -1 : Non renseigné
- 0 : Non renseigné ou sans objet
- 1 : Se déplaçant dans le Sens véhicule heurtant

	<ul style="list-style-type: none"> <li>- 2 : Se déplaçant dans le Sens inverse du véhicule</li> <li>- 3 : Traversant</li> <li>- 4 : Masqué</li> <li>- 5 : Jouant – courant</li> <li>- 6 : Avec animal</li> <li>- 9 : Autre</li> <li>- A : Monte/descend du véhicule</li> <li>- B : Inconnue</li> </ul>										
Type	[2005-2008] : int64 [2009-2018] : float64 [2019-2022] : object										
Etendue des valeurs	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">count</th><th style="text-align: center;">unique</th><th style="text-align: center;">top</th><th style="text-align: center;">freq</th></tr> </thead> <tbody> <tr> <td><b>actp</b></td><td style="text-align: center;">2579915</td><td style="text-align: center;">21</td><td style="text-align: center;">0.0</td><td style="text-align: center;">1224776</td></tr> </tbody> </table>		count	unique	top	freq	<b>actp</b>	2579915	21	0.0	1224776
	count	unique	top	freq							
<b>actp</b>	2579915	21	0.0	1224776							
Valeurs nulles	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">Type</th><th style="text-align: center;">Val_notnull</th><th style="text-align: center;">Val_null</th><th style="text-align: center;">%_null</th></tr> </thead> <tbody> <tr> <td><b>actp</b></td><td style="text-align: center;">object</td><td style="text-align: center;">2579915</td><td style="text-align: center;">56462</td><td style="text-align: center;">2.14</td></tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>actp</b>	object	2579915	56462	2.14
	Type	Val_notnull	Val_null	%_null							
<b>actp</b>	object	2579915	56462	2.14							
Outliers	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">outliers_count</th><th style="text-align: center;">outliers_unique</th><th style="text-align: center;">outliers_list</th></tr> </thead> <tbody> <tr> <td><b>actp</b></td><td style="text-align: center;">50975</td><td style="text-align: center;">16</td><td style="text-align: center;">[ 1, 1.0, 2, 2.0, 4, 4.0, 5, 5.0, 6, 6.0, 7, 8...</td></tr> </tbody> </table> <p style="text-align: center;">Boxplots for column: actp</p>		outliers_count	outliers_unique	outliers_list	<b>actp</b>	50975	16	[ 1, 1.0, 2, 2.0, 4, 4.0, 5, 5.0, 6, 6.0, 7, 8...		
	outliers_count	outliers_unique	outliers_list								
<b>actp</b>	50975	16	[ 1, 1.0, 2, 2.0, 4, 4.0, 5, 5.0, 6, 6.0, 7, 8...								

Répartition			
	Count	% valeurs	
<b>Modalité</b>			
-1	185666	7.0	
0	959099	37.0	
0.0	1224776	47.0	
1	5470	0.0	
1.0	6758	0.0	
2	2775	0.0	
2.0	3239	0.0	
3	70470	3.0	
3.0	88929	3.0	
4	1378	0.0	
4.0	2389	0.0	
5	4737	0.0	
5.0	7196	0.0	
6	239	0.0	
6.0	275	0.0	
7	60	0.0	
8	52	0.0	
9	6580	0.0	
9.0	8171	0.0	
A	422	0.0	
B	1234	0.0	
<b>Remarque</b>			Attention aux modalités -1, 0 pour lesquelles il pourrait y avoir un amalgame. Les valeurs NaN peuvent être remplacées par -1 qui signifie « non renseigné ».

### j. etatp

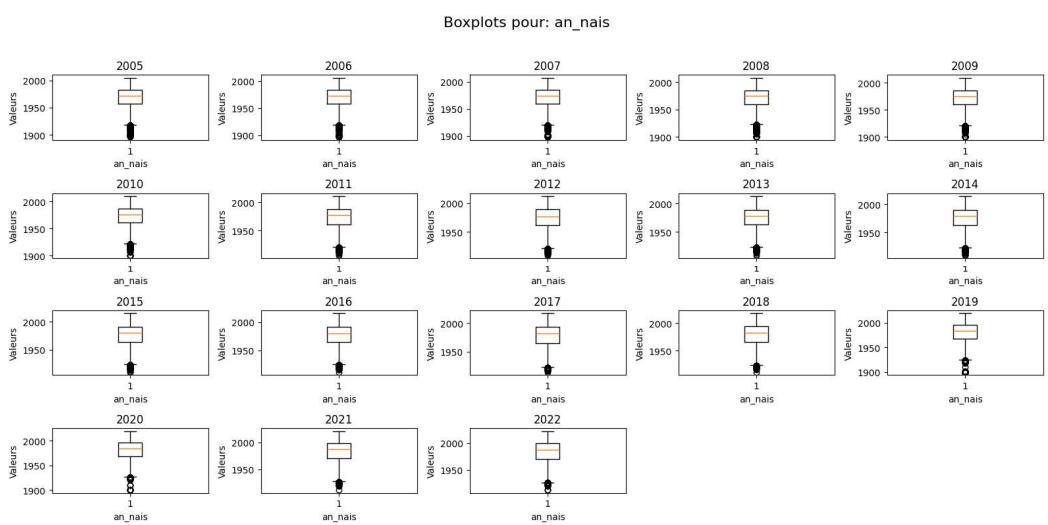
Description	Cette variable permet de préciser si le piéton accidenté était seul ou non.
Modalités	<ul style="list-style-type: none"> <li>- -1 : Non renseigné</li> <li>- 1 : Seul</li> <li>- 2 : Accompagné</li> <li>- 3 : En groupe</li> </ul>
Type	[2005-2008 ; 2019-2022] : int64 [2009-2018] : float64

<b>Etendue des valeurs</b>	<b>count</b>	<b>unique</b>	<b>top</b>	<b>freq</b>
	<b>etatp</b>	2579959	5	0.0
<b>Valeurs nulles</b>	<b>Type</b>	<b>Val_notnull</b>	<b>Val_null</b>	<b>%_null</b>
	<b>etatp</b>	float64	2579959	56418
<b>Outliers</b>	<b>outliers_count</b>	<b>outliers_unique</b>	<b>outliers_list</b>	
	<b>etatp</b>	665166	4	[ -1.0, 1.0, 2.0, 3.0]
Boxplots for column: etatp				
<b>Répartition</b>	<b>Count</b>	<b>% valeurs</b>		
	<b>Modalité</b>			
-1.0	456246	18.0		
0.0	1914793	74.0		
1.0	158481	6.0		
2.0	41509	2.0		
3.0	8930	0.0		

<b>Evolution</b>	<p>Evolution de la distribution etatp</p> <p>Proportion (%)</p> <p>Années</p> <p>Modalités</p> <ul style="list-style-type: none"> <li>0</li> <li>1</li> <li>2</li> <li>3</li> <li>-1</li> </ul>
<b>Remarque</b>	À partir de 2019, la modalité -1 semble remplacer 0. Les valeurs NaN peuvent être remplacées par -1 qui signifie « non renseigné ».

### k. an\_nais

<b>Description</b>	Année de naissance de l'usager.										
<b>Type</b>	[2005-2018 ; 2021-2022] : float64 [2019-2020] : int64										
<b>Etendue des valeurs</b>	<table> <thead> <tr> <th></th> <th>count</th> <th>unique</th> <th>top</th> <th>freq</th> </tr> </thead> <tbody> <tr> <td><b>an_nais</b></td> <td>2628018</td> <td>127</td> <td>1988.0</td> <td>66282</td> </tr> </tbody> </table>		count	unique	top	freq	<b>an_nais</b>	2628018	127	1988.0	66282
	count	unique	top	freq							
<b>an_nais</b>	2628018	127	1988.0	66282							
<b>Valeurs nulles</b>	<table> <thead> <tr> <th></th> <th>Type</th> <th>Val_notnull</th> <th>Val_null</th> <th>%_null</th> </tr> </thead> <tbody> <tr> <td><b>an_nais</b></td> <td>float64</td> <td>2628018</td> <td>8359</td> <td>0.32</td> </tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>an_nais</b>	float64	2628018	8359	0.32
	Type	Val_notnull	Val_null	%_null							
<b>an_nais</b>	float64	2628018	8359	0.32							
<b>Outliers</b>	<table> <thead> <tr> <th></th> <th>outliers_count</th> <th>outliers_unique</th> <th>outliers_list</th> </tr> </thead> <tbody> <tr> <td><b>an_nais</b></td> <td>13290</td> <td>28</td> <td>[ 1896.0, 1897.0, 1898.0, 1899.0, 1900.0, 1901...]</td> </tr> </tbody> </table>		outliers_count	outliers_unique	outliers_list	<b>an_nais</b>	13290	28	[ 1896.0, 1897.0, 1898.0, 1899.0, 1900.0, 1901...]		
	outliers_count	outliers_unique	outliers_list								
<b>an_nais</b>	13290	28	[ 1896.0, 1897.0, 1898.0, 1899.0, 1900.0, 1901...]								



## Répartition

Count % valeurs

### an\_nais

<b>1896.0</b>	1	0.0
<b>1897.0</b>	3	0.0
<b>1898.0</b>	35	0.0
<b>1899.0</b>	2	0.0
<b>1900.0</b>	286	0.0
...	...	...
<b>2019.0</b>	1179	0.0
<b>2020.0</b>	758	0.0
<b>2021.0</b>	525	0.0
<b>2022.0</b>	198	0.0
<b>NaN</b>	8359	0.0

128 rows × 2 columns

## Remarque

Attention aux outliers très bas, probablement le format de saisie (pas la date entière).

## I. num\_veh

Description	Identifiant du véhicule repris pour chacun des usagers occupant ce véhicule - code alphanumérique.
Type	object

<b>Etendue des valeurs</b>	<b>count</b>	<b>unique</b>	<b>top</b>	<b>freq</b>
	<b>num_veh</b>	2636377	181	A01 1601497
<b>Valeurs nulles</b>	<b>Type</b>	<b>Val_notnull</b>	<b>Val_null</b>	<b>%_null</b>
	<b>num_veh</b>	object	2636377	0 0.0
<b>Outliers</b>	<b>outliers_count</b>	<b>outliers_unique</b>		
	<b>num_veh</b>	63930	177	[ A02, A03, A04, A05, A06, A07, A08, A09, A27,...
<b>Répartition</b>	<b>Count</b>	<b>% valeurs</b>		
	<b>num_veh</b>			
	<b>A01</b>	1601497	61.0	
	<b>A02</b>	589	0.0	
	<b>A03</b>	49	0.0	
	<b>A04</b>	7	0.0	
	<b>A05</b>	5	0.0	
	<b>...</b>	...	...	
	<b>Z01</b>	1	0.0	
	<b>ZZ01</b>	5	0.0	
181 rows × 2 columns				

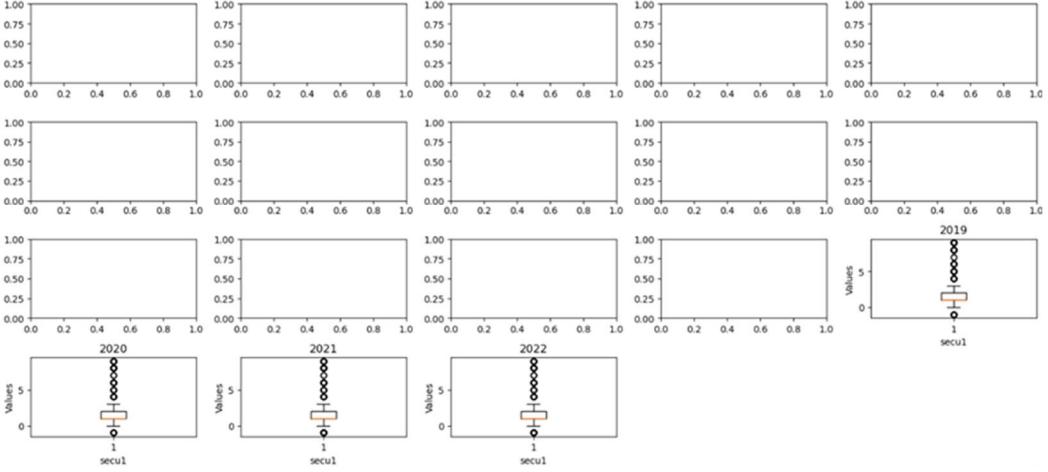
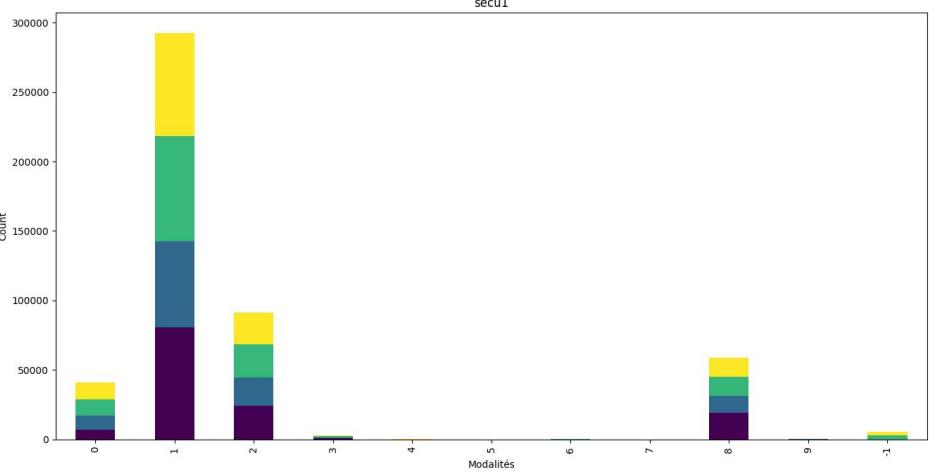
### m.id\_véhicule

<b>Description</b>	Identifiant unique du véhicule repris pour chacun des usagers occupant ce véhicule - code numérique.
<b>Type</b>	[2019-2022] : object
<b>Etendue des valeurs</b>	<b>count</b>
	<b>unique</b>
<b>Valeurs nulles</b>	<b>top</b>
	<b>freq</b>
<b>Valeurs nulles</b>	<b>Type</b>
	<b>Val_notnull</b>
<b>Valeurs nulles</b>	<b>Val_null</b>
	<b>%_null</b>
<b>Valeurs nulles</b>	<b>id_véhicule</b>
	object
<b>Valeurs nulles</b>	494182
	2142195
<b>Valeurs nulles</b>	81.26

Outliers	outliers_count outliers_unique			outliers_list
	id_vehicule	494182	369639	
Répartition	<b>Count % valeurs</b>			
	<b>id_vehicule</b>			
	<b>100 882</b>	1	0.0	
	<b>100 883</b>	1	0.0	
	<b>100 884</b>	1	0.0	
	<b>100 885</b>	1	0.0	
	<b>100 886</b>	1	0.0	
	...	...	...	
	<b>813 950</b>	1	0.0	
	<b>813 951</b>	1	0.0	
	<b>813 952</b>	1	0.0	
	<b>813 953</b>	1	0.0	
	<b>NaN</b>	2142195	81.0	
	369640 rows × 2 columns			

## n. secu1

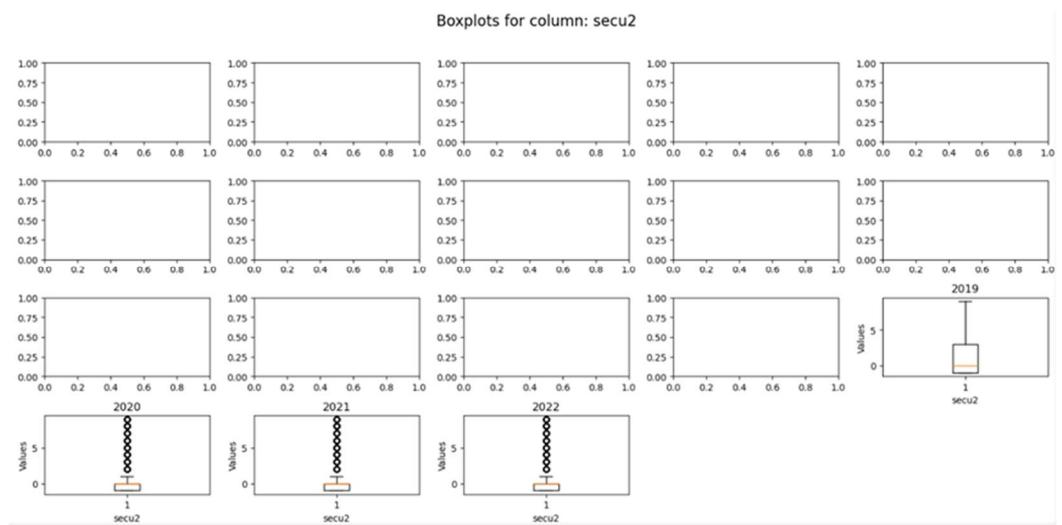
Description	Le renseignement du caractère indique la présence et l'utilisation de l'équipement de sécurité.										
Modalités	<ul style="list-style-type: none"> <li>- -1 : Non renseigné</li> <li>- 0 : Aucun équipement</li> <li>- 1 : Ceinture</li> <li>- 2 : Casque</li> <li>- 3 : Dispositif enfants</li> <li>- 4 : Gilet réfléchissant</li> <li>- 5 : Airbag (2RM/3RM)</li> <li>- 6 : Gants (2RM/3RM)</li> <li>- 7 : Gants + Airbag (2RM/3RM)</li> <li>- 8 : Non déterminable</li> <li>- 9 : Autre</li> </ul>										
Type	[2019-2022] : int64										
Etendue des valeurs	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th></th><th>count</th><th>unique</th><th>top</th><th>freq</th></tr> </thead> <tbody> <tr> <td><b>secu1</b></td><td>494182</td><td>11</td><td>1.0</td><td>292332</td></tr> </tbody> </table>		count	unique	top	freq	<b>secu1</b>	494182	11	1.0	292332
	count	unique	top	freq							
<b>secu1</b>	494182	11	1.0	292332							

<b>Valeurs nulles</b>	<b>Type</b>	<b>Val_notnull</b>	<b>Val_null</b>	<b>%_null</b>
	<b>secu1</b>	float64	494182	2142195 81.26
<b>Outliers</b>	<b>outliers_count</b>	<b>outliers_unique</b>	<b>outliers_list</b>	
	<b>secu1</b>	66208	7	[ -1.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0]
Boxplots for column: secu1				
				
<b>Répartition</b>	<b>Count</b>	<b>% valeurs</b>		
	<b>Modalité</b>			
-1.0	5591	1.0		
0.0	41182	8.0		
1.0	292332	59.0		
2.0	91302	18.0		
3.0	3158	1.0		
4.0	334	0.0		
5.0	219	0.0		
6.0	399	0.0		
7.0	15	0.0		
8.0	59115	12.0		
9.0	535	0.0		
				

<b>Evolution</b>	<p style="text-align: center;">Evolution de la distribution secu1</p>
<b>Remarque</b>	Les valeurs NaN peuvent être remplacées par -1 qui signifie « non renseigné ».

## o. secu2

<b>Description</b>	Le renseignement du caractère indique la présence et l'utilisation de l'équipement de sécurité.										
<b>Modalités</b>	<ul style="list-style-type: none"> <li>- -1 : Non renseigné</li> <li>- 0 : Aucun équipement</li> <li>- 1 : Ceinture</li> <li>- 2 : Casque</li> <li>- 3 : Dispositif enfants</li> <li>- 4 : Gilet réfléchissant</li> <li>- 5 : Airbag (2RM/3RM)</li> <li>- 6 : Gants (2RM/3RM)</li> <li>- 7 : Gants + Airbag (2RM/3RM)</li> <li>- 8 : Non déterminable</li> <li>- 9 : Autre</li> </ul>										
<b>Type</b>	int64										
<b>Etendue des valeurs</b>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: left;">count</th> <th style="text-align: left;">unique</th> <th style="text-align: left;">top</th> <th style="text-align: left;">freq</th> </tr> </thead> <tbody> <tr> <td><b>secu2</b></td> <td>494182</td> <td>11</td> <td>-1.0</td> <td>193509</td> </tr> </tbody> </table>		count	unique	top	freq	<b>secu2</b>	494182	11	-1.0	193509
	count	unique	top	freq							
<b>secu2</b>	494182	11	-1.0	193509							
<b>Valeurs nulles</b>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: left;">Type</th> <th style="text-align: left;">Val_notnull</th> <th style="text-align: left;">Val_null</th> <th style="text-align: left;">%_null</th> </tr> </thead> <tbody> <tr> <td><b>secu2</b></td> <td>float64</td> <td>494182</td> <td>2142195</td> <td>81.26</td> </tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>secu2</b>	float64	494182	2142195	81.26
	Type	Val_notnull	Val_null	%_null							
<b>secu2</b>	float64	494182	2142195	81.26							
<b>Outliers</b>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: left;">outliers_count</th> <th style="text-align: left;">outliers_unique</th> <th style="text-align: right;">outliers_list</th> </tr> </thead> <tbody> <tr> <td><b>secu2</b></td> <td>109988</td> <td>8</td> <td style="text-align: right;">[ 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0]</td> </tr> </tbody> </table>		outliers_count	outliers_unique	outliers_list	<b>secu2</b>	109988	8	[ 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0]		
	outliers_count	outliers_unique	outliers_list								
<b>secu2</b>	109988	8	[ 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0]								

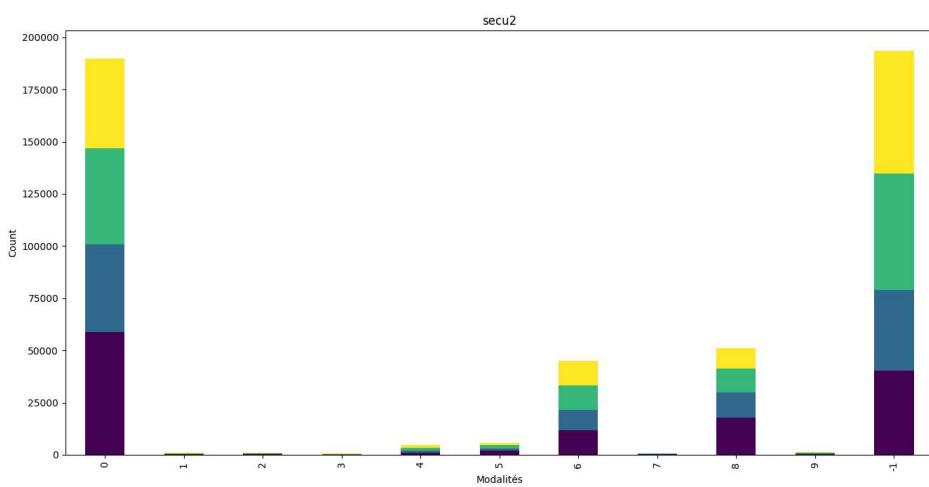


## Répartition

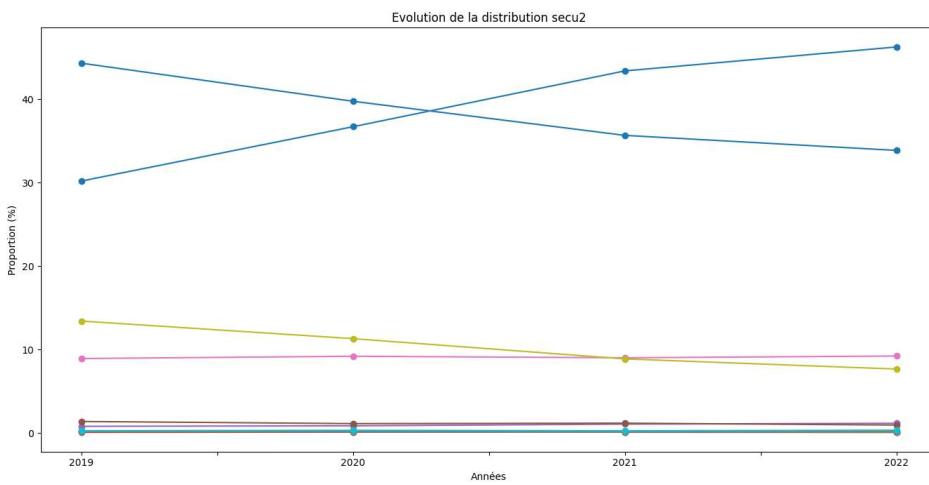
Count % valeurs

### Modalité

-1.0	193509	39.0
0.0	189789	38.0
1.0	896	0.0
2.0	841	0.0
3.0	555	0.0
4.0	4845	1.0
5.0	5771	1.0
6.0	44898	9.0
7.0	656	0.0
8.0	50941	10.0
9.0	1481	0.0



## Evolution



## Remarque

Les valeurs NaN peuvent être remplacées par -1 qui signifie « non renseigné ».

## p. secu3

Description	Le renseignement du caractère indique la présence et l'utilisation de l'équipement de sécurité :										
Modalités	<ul style="list-style-type: none"> <li>- -1 : Non renseigné</li> <li>- 0 : Aucun équipement</li> <li>- 1 : Ceinture</li> <li>- 2 : Casque</li> <li>- 3 : Dispositif enfants</li> <li>- 4 : Gilet réfléchissant</li> <li>- 5 : Airbag (2RM/3RM)</li> <li>- 6 : Gants (2RM/3RM)</li> <li>- 7 : Gants + Airbag (2RM/3RM)</li> <li>- 8 : Non déterminable</li> <li>- 9 : Autre</li> </ul>										
Type	int64										
Etendue des valeurs	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">count</th><th style="text-align: center;">unique</th><th style="text-align: center;">top</th><th style="text-align: center;">freq</th></tr> </thead> <tbody> <tr> <td><b>secu3</b></td><td style="text-align: center;">494182</td><td style="text-align: center;">11</td><td style="text-align: center;">-1.0</td><td style="text-align: center;">488588</td></tr> </tbody> </table>		count	unique	top	freq	<b>secu3</b>	494182	11	-1.0	488588
	count	unique	top	freq							
<b>secu3</b>	494182	11	-1.0	488588							
Valeurs nulles	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">Type</th><th style="text-align: center;">Val_notnull</th><th style="text-align: center;">Val_null</th><th style="text-align: center;">%_null</th></tr> </thead> <tbody> <tr> <td><b>secu3</b></td><td style="text-align: center;">float64</td><td style="text-align: center;">494182</td><td style="text-align: center;">2142195</td><td style="text-align: center;">81.26</td></tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>secu3</b>	float64	494182	2142195	81.26
	Type	Val_notnull	Val_null	%_null							
<b>secu3</b>	float64	494182	2142195	81.26							
Outliers	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">outliers_count</th><th style="text-align: center;">outliers_unique</th><th style="text-align: center;">outliers_list</th></tr> </thead> <tbody> <tr> <td><b>secu3</b></td><td style="text-align: center;">5594</td><td style="text-align: center;">10</td><td style="text-align: center;">[ 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0,...</td></tr> </tbody> </table> <p style="text-align: center;">Boxplots for column: secu3</p>		outliers_count	outliers_unique	outliers_list	<b>secu3</b>	5594	10	[ 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0,...		
	outliers_count	outliers_unique	outliers_list								
<b>secu3</b>	5594	10	[ 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0,...								

<h3>Répartition</h3> <table border="1"> <thead> <tr> <th></th><th>Count</th><th>% valeurs</th></tr> </thead> <tbody> <tr><td>Modalité</td><td></td><td></td></tr> <tr><td>-1.0</td><td>488588</td><td>99.0</td></tr> <tr><td>0.0</td><td>1330</td><td>0.0</td></tr> <tr><td>1.0</td><td>87</td><td>0.0</td></tr> <tr><td>2.0</td><td>15</td><td>0.0</td></tr> <tr><td>3.0</td><td>7</td><td>0.0</td></tr> <tr><td>4.0</td><td>71</td><td>0.0</td></tr> <tr><td>5.0</td><td>38</td><td>0.0</td></tr> <tr><td>6.0</td><td>250</td><td>0.0</td></tr> <tr><td>7.0</td><td>12</td><td>0.0</td></tr> <tr><td>8.0</td><td>235</td><td>0.0</td></tr> <tr><td>9.0</td><td>3549</td><td>1.0</td></tr> </tbody> </table>		Count	% valeurs	Modalité			-1.0	488588	99.0	0.0	1330	0.0	1.0	87	0.0	2.0	15	0.0	3.0	7	0.0	4.0	71	0.0	5.0	38	0.0	6.0	250	0.0	7.0	12	0.0	8.0	235	0.0	9.0	3549	1.0	
	Count	% valeurs																																						
Modalité																																								
-1.0	488588	99.0																																						
0.0	1330	0.0																																						
1.0	87	0.0																																						
2.0	15	0.0																																						
3.0	7	0.0																																						
4.0	71	0.0																																						
5.0	38	0.0																																						
6.0	250	0.0																																						
7.0	12	0.0																																						
8.0	235	0.0																																						
9.0	3549	1.0																																						
<h3>Evolution</h3>																																								
<b>Remarque</b>	Les valeurs NaN peuvent être remplacées par -1 qui signifie « non renseigné ».																																							

## q. id\_usager

<b>Description</b>	Identifiant unique de l'usager - code numérique.										
<b>Type</b>	object										
<b>Etendue des valeurs</b>	<table border="1"> <thead> <tr> <th></th> <th>count</th> <th>unique</th> <th>top</th> <th>freq</th> </tr> </thead> <tbody> <tr> <td><b>id_usager</b></td> <td>255910</td> <td>255910</td> <td>133 818</td> <td>1</td> </tr> </tbody> </table>		count	unique	top	freq	<b>id_usager</b>	255910	255910	133 818	1
	count	unique	top	freq							
<b>id_usager</b>	255910	255910	133 818	1							
<b>Valeurs nulles</b>	<table border="1"> <thead> <tr> <th></th> <th>Type</th> <th>Val_notnull</th> <th>Val_null</th> <th>%_null</th> </tr> </thead> <tbody> <tr> <td><b>id_usager</b></td> <td>object</td> <td>255910</td> <td>2380467</td> <td>90.29</td> </tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>id_usager</b>	object	255910	2380467	90.29
	Type	Val_notnull	Val_null	%_null							
<b>id_usager</b>	object	255910	2380467	90.29							

Outliers	outliers_count outliers_unique			outliers_list
	id_usager	255910	255910 [ 133 818, 133 819, 133 820, 133 821, 133 822,...	
Répartition	Count % valeurs			
	id_usager			
	133 818	1	0.0	
	133 819	1	0.0	
	133 820	1	0.0	
	133 821	1	0.0	
	133 822	1	0.0	
	...	...	...	
	999 994	1	0.0	
	999 996	1	0.0	
	999 998	1	0.0	
	999 999	1	0.0	
NaN 2380467 90.0				
255911 rows × 2 columns				

## 4. Véhicules

Rows x columns Rows duplicated

Véhicules (2009395, 11) 0

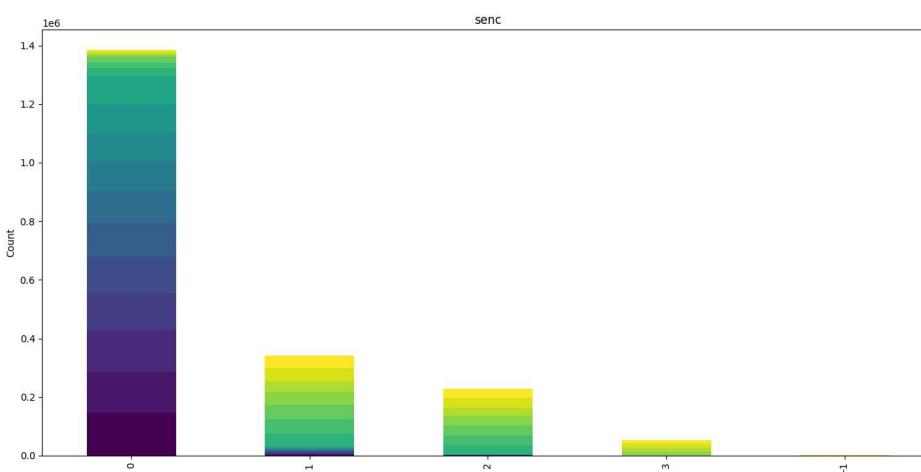
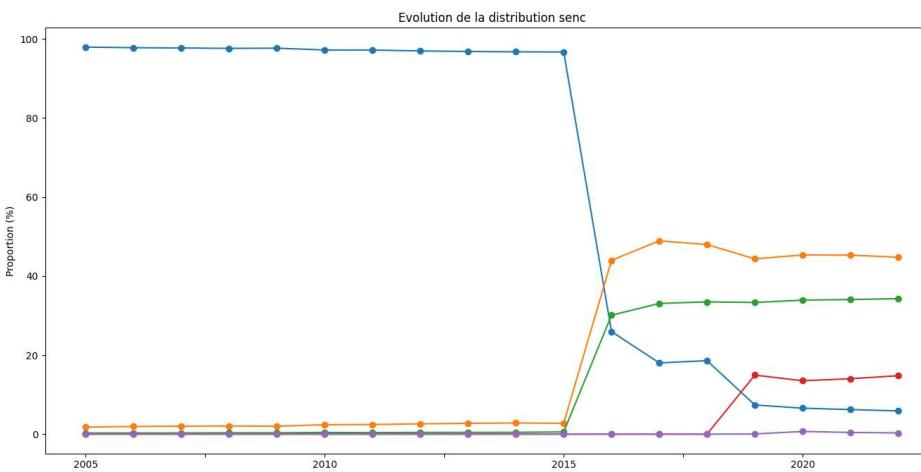
### a. Num\_Acc

Description	Identifiant de l'accident identique à celui du fichier "rubrique CARACTERISTIQUES" repris pour chacun des véhicules décrits impliqués dans l'accident.
Type	int64
Etendue des valeurs	count unique top freq
	Num_Acc 2009395 1176873 200600074917 56
Valeurs nulles	Type Val_notnull Val_null %_null
	Num_Acc int64 2009395 0 0.0

Outliers	outliers_count outliers_unique outliers_list		
	Num_Acc	0	0
Répartition	Count % valeurs		
	Num_Acc		
	200500000001	2	0.0
	200500000002	2	0.0
	200500000003	2	0.0
	200500000004	3	0.0
	200500000005	1	0.0
	...	...	...
	202200055298	1	0.0
	202200055299	1	0.0
	202200055300	1	0.0
	202200055301	2	0.0
	202200055302	2	0.0
	1176873 rows × 2 columns		

### b. senc

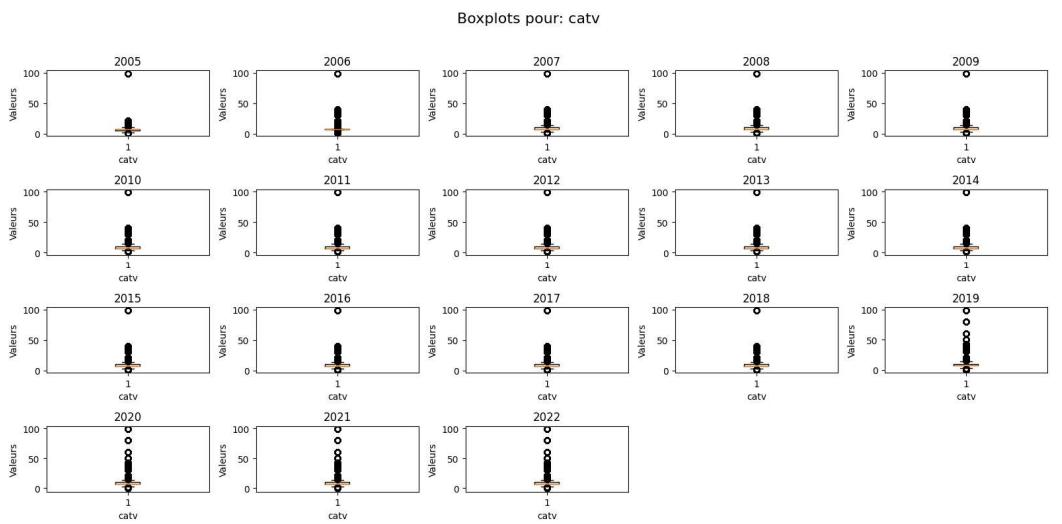
Description	Sens de circulation.
Modalités	- -1 : Non renseigné - 0 : Inconnu - 1 : PK ou PR ou numéro d'adresse postale croissant - 2 : PK ou PR ou numéro d'adresse postale décroissant - 3 : Absence de repère
Type	[2005-2015 ; 2019-2022] : int64 [2016-2018] : float64
Etendue des valeurs	count unique top freq
	senc 2009123 5 0.0 1384153
Valeurs nulles	Type Val_notnull Val_null %_null
	senc float64 2009123 272 0.01

Outliers	outliers_count outliers_unique outliers_list																																	
	senc	53593	1 [ 3.0 ]																															
<b>Répartition</b>		 <p>A stacked bar chart titled "Répartition" showing the count of senc values from 0 to 3 across years from 2005 to 2022. The total count is 53593. The x-axis represents Modalités (0, 1, 2, 3) and the y-axis represents count (0.0 to 1.4e6). A color scale legend on the right indicates the years from 2005 to 2022, with darker shades representing earlier years.</p> <table border="1"> <thead> <tr> <th>Modalité</th> <th>Count</th> <th>% valeurs</th> </tr> </thead> <tbody> <tr> <td>-1.0</td> <td>1287</td> <td>0.0</td> </tr> <tr> <td>0.0</td> <td>1384153</td> <td>69.0</td> </tr> <tr> <td>1.0</td> <td>340696</td> <td>17.0</td> </tr> <tr> <td>2.0</td> <td>229394</td> <td>11.0</td> </tr> <tr> <td>3.0</td> <td>53593</td> <td>3.0</td> </tr> <tr> <td>NaN</td> <td>272</td> <td>0.0</td> </tr> </tbody> </table>			Modalité	Count	% valeurs	-1.0	1287	0.0	0.0	1384153	69.0	1.0	340696	17.0	2.0	229394	11.0	3.0	53593	3.0	NaN	272	0.0									
Modalité	Count	% valeurs																																
-1.0	1287	0.0																																
0.0	1384153	69.0																																
1.0	340696	17.0																																
2.0	229394	11.0																																
3.0	53593	3.0																																
NaN	272	0.0																																
<b>Evolution</b>		 <p>A line chart titled "Evolution de la distribution senc" showing the proportion (%) of senc values from 0 to 3 from 2005 to 2022. The proportion for modalité 0 remained at 100% until 2015, then dropped sharply to around 10%. Modalité 1 increased from 0% to about 45%. Modalité 2 increased from 0% to about 35%. Modalité 3 increased from 0% to about 15%.</p> <table border="1"> <thead> <tr> <th>Années</th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> </tr> </thead> <tbody> <tr> <td>2005</td> <td>100</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>2010</td> <td>100</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>2015</td> <td>100</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>2020</td> <td>10</td> <td>45</td> <td>35</td> <td>15</td> </tr> <tr> <td>2022</td> <td>10</td> <td>45</td> <td>35</td> <td>15</td> </tr> </tbody> </table>			Années	0	1	2	3	2005	100	0	0	0	2010	100	0	0	0	2015	100	0	0	0	2020	10	45	35	15	2022	10	45	35	15
Années	0	1	2	3																														
2005	100	0	0	0																														
2010	100	0	0	0																														
2015	100	0	0	0																														
2020	10	45	35	15																														
2022	10	45	35	15																														
Remarque	Les valeurs NaN peuvent être remplacées par -1 qui signifie « non renseigné ».																																	

### c. catv

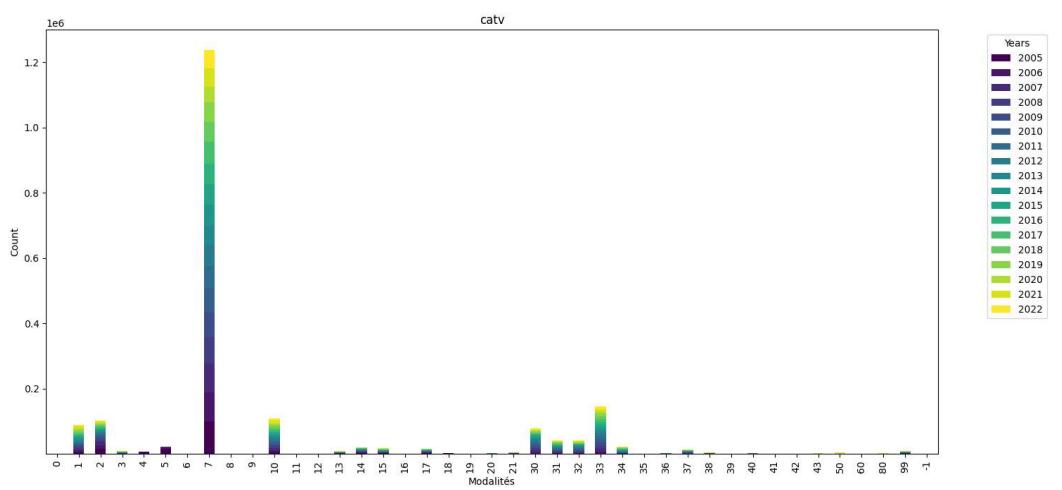
Description	Catégorie du véhicule.
Modalités	<ul style="list-style-type: none"> <li>- 00 : Indéterminable</li> <li>- 01 : Bicyclette</li> <li>- 02 : Cyclomoteur &lt;50cm3</li> <li>- 03 : Voiturette (Quadricycle à moteur carrossé)</li> <li>- 04 : Référence inutilisée depuis 2006 (scooter immatriculé)</li> <li>- 05 : Référence inutilisée depuis 2006 (motocyclette)</li> <li>- 06 : Référence inutilisée depuis 2006 (side-car)</li> <li>- 07 : VL seul</li> <li>- 08 : Référence inutilisée depuis 2006 (VL + caravane)</li> <li>- 09 : Référence inutilisée depuis 2006 (VL + remorque)</li> <li>- 10 : VU seul 1,5T &lt;= PTAC &lt;= 3,5T</li> <li>- 11 : Référence inutilisée depuis 2006 (VU (10) + caravane)</li> </ul>

	- 12 : Référence inutilisée depuis 2006 (VU (10) + remorque) - 13 : PL seul 3,5T <PTCA <= 7,5T - 14 : PL seul > 7,5T - 15 : PL > 3,5T + remorque - 16 : Tracteur routier seul - 17 : Tracteur routier + semi-remorque - 18 : Référence inutilisée depuis 2006 (transport en commun) - 19 : Référence inutilisée depuis 2006 (tramway) - 20 : Engin spécial - 21 : Tracteur agricole - 30 : Scooter < 50 cm3 - 31 : Motocyclette > 50 cm3 et <= 125 cm3 - 32 : Scooter > 50 cm3 et <= 125 cm3 - 33 : Motocyclette > 125 cm3 - 34 : Scooter > 125 cm3 - 35 : Quad léger <= 50 cm3 (Quadricycle à moteur non carrossé) - 36 : Quad lourd > 50 cm3 (Quadricycle à moteur non carrossé) - 37 : Autobus - 38 : Autocar - 39 : Train - 40 : Tramway - 41 : 3RM <= 50 cm3 - 42 : 3RM > 50 cm3 <= 125 cm3 - 43 : 3RM > 125 cm3 - 50 : EDP à moteur - 60 : EDP sans moteur - 80 : VAE - 99 : Autre véhicule										
Type	int64										
Etendue des valeurs	<table> <thead> <tr> <th></th><th>count</th><th>unique</th><th>top</th><th>freq</th></tr> </thead> <tbody> <tr> <td><b>catv</b></td><td>2009395</td><td>41</td><td>7</td><td>1237634</td></tr> </tbody> </table>		count	unique	top	freq	<b>catv</b>	2009395	41	7	1237634
	count	unique	top	freq							
<b>catv</b>	2009395	41	7	1237634							
Valeurs nulles	<table> <thead> <tr> <th></th><th>Type</th><th>Val_notnull</th><th>Val_null</th><th>%_null</th></tr> </thead> <tbody> <tr> <td><b>catv</b></td><td>int64</td><td>2009395</td><td>0</td><td>0.0</td></tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>catv</b>	int64	2009395	0	0.0
	Type	Val_notnull	Val_null	%_null							
<b>catv</b>	int64	2009395	0	0.0							
Outliers	<table> <thead> <tr> <th></th><th>outliers_count</th><th>outliers_unique</th><th>outliers_list</th></tr> </thead> <tbody> <tr> <td><b>catv</b></td><td>599066</td><td>29</td><td>[ -1.0, 0.0, 1.0, 2.0, 15.0, 16.0, 17.0, 18.0, ...]</td></tr> </tbody> </table>		outliers_count	outliers_unique	outliers_list	<b>catv</b>	599066	29	[ -1.0, 0.0, 1.0, 2.0, 15.0, 16.0, 17.0, 18.0, ...]		
	outliers_count	outliers_unique	outliers_list								
<b>catv</b>	599066	29	[ -1.0, 0.0, 1.0, 2.0, 15.0, 16.0, 17.0, 18.0, ...]								



### Répartition

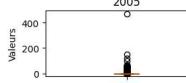
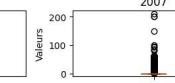
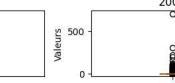
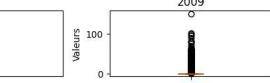
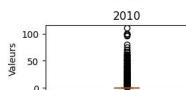
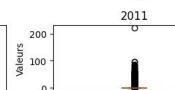
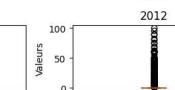
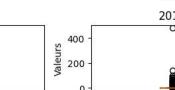
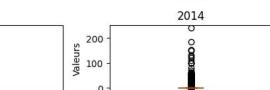
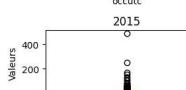
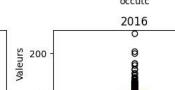
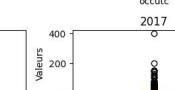
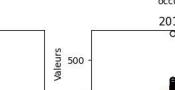
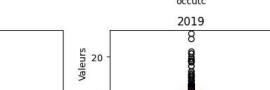
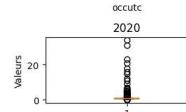
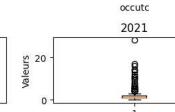
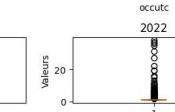
	Count	% valeurs
<b>catv</b>		
<b>-1</b>	13	0.0
<b>0</b>	1030	0.05
<b>1</b>	88885	4.42
<b>2</b>	101713	5.06
<b>3</b>	8072	0.4
...	...	...
<b>43</b>	1995	0.1
<b>50</b>	5116	0.25
<b>60</b>	754	0.04
<b>80</b>	1793	0.09
<b>99</b>	8283	0.41
41 rows × 2 columns		



**Remarque** On a une apparition de plus de modalités à partir de 2019.

### d. occutc

Description	Nombre d'occupants dans le transport en commun.
Type	[2005-2018] : int64 [2019-2022] : float64

<b>Etendue des valeurs</b>		<b>count</b>	<b>unique</b>	<b>top</b>	<b>freq</b>
	<b>occutc</b>	1638885	124	0.0	1624683
<b>Valeurs nulles</b>		<b>Type</b>	<b>Val_notnull</b>	<b>Val_null</b>	<b>%_null</b>
	<b>occutc</b>	float64	1638885	370510	18.44
<b>Outliers</b>		<b>outliers_count</b>	<b>outliers_unique</b>		<b>outliers_list</b>
	<b>occutc</b>	14202	123	[ 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0,...	
	Boxplots pour: occutc				
					
					
					
					
<b>Répartition</b>		<b>Count</b>	<b>% valeurs</b>		
	<b>occutc</b>				
	<b>0.0</b>	1624683	81.0		
	<b>1.0</b>	7699	0.0		
	<b>2.0</b>	1127	0.0		
	<b>3.0</b>	581	0.0		
	<b>4.0</b>	301	0.0		
	<b>...</b>	<b>...</b>	<b>...</b>		
	<b>480.0</b>	1	0.0		
	<b>490.0</b>	1	0.0		
	<b>700.0</b>	1	0.0		
	<b>900.0</b>	1	0.0		
	<b>NaN</b>	370510	18.0		
	125 rows × 2 columns				

<b>Remarque</b>	Beaucoup de valeurs manquantes et certaines valeurs sont aberrantes (trop de passagers).
-----------------	--

### e. obs

<b>Description</b>	Obstacle fixe heurté.										
<b>Modalités</b>	<ul style="list-style-type: none"> <li>- -1 : Non renseigné</li> <li>- 0 : Sans objet</li> <li>- 1 : Véhicule en stationnement</li> <li>- 2 : Arbre</li> <li>- 3 : Glissière métallique</li> <li>- 4 : Glissière béton</li> <li>- 5 : Autre glissière</li> <li>- 6 : Bâtiment, mur, pile de pont</li> <li>- 7 : Support de signalisation verticale ou poste d'appel d'urgence</li> <li>- 8 : Poteau</li> <li>- 9 : Mobilier urbain</li> <li>- 10 : Parapet</li> <li>- 11 : Ilot, refuge, borne haute</li> <li>- 12 : Bordure de trottoir</li> <li>- 13 : Fossé, talus, paroi rocheuse</li> <li>- 14 : Autre obstacle fixe sur chaussée</li> <li>- 15 : Autre obstacle fixe sur trottoir ou accotement</li> <li>- 16 : Sortie de chaussée sans obstacle</li> <li>- 17 : Buse - tête d'aqueduc</li> </ul>										
<b>Type</b>	[2005-2008 ; 2019-2022] : int64 [2009-2018] : float64										
<b>Etendue des valeurs</b>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: center;"><b>count</b></th> <th style="text-align: center;"><b>unique</b></th> <th style="text-align: center;"><b>top</b></th> <th style="text-align: center;"><b>freq</b></th> </tr> </thead> <tbody> <tr> <td><b>obs</b></td> <td style="text-align: center;">2008389</td> <td style="text-align: center;">19</td> <td style="text-align: center;">0.0</td> <td style="text-align: center;">1740874</td> </tr> </tbody> </table>		<b>count</b>	<b>unique</b>	<b>top</b>	<b>freq</b>	<b>obs</b>	2008389	19	0.0	1740874
	<b>count</b>	<b>unique</b>	<b>top</b>	<b>freq</b>							
<b>obs</b>	2008389	19	0.0	1740874							
<b>Valeurs nulles</b>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: center;"><b>Type</b></th> <th style="text-align: center;"><b>Val_notnull</b></th> <th style="text-align: center;"><b>Val_null</b></th> <th style="text-align: center;"><b>%_null</b></th> </tr> </thead> <tbody> <tr> <td><b>obs</b></td> <td style="text-align: center;">float64</td> <td style="text-align: center;">2008389</td> <td style="text-align: center;">1006</td> <td style="text-align: center;">0.05</td> </tr> </tbody> </table>		<b>Type</b>	<b>Val_notnull</b>	<b>Val_null</b>	<b>%_null</b>	<b>obs</b>	float64	2008389	1006	0.05
	<b>Type</b>	<b>Val_notnull</b>	<b>Val_null</b>	<b>%_null</b>							
<b>obs</b>	float64	2008389	1006	0.05							
<b>Outliers</b>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: center;"><b>outliers_count</b></th> <th style="text-align: center;"><b>outliers_unique</b></th> <th style="text-align: center;"><b>outliers_list</b></th> </tr> </thead> <tbody> <tr> <td><b>obs</b></td> <td style="text-align: center;">267515</td> <td style="text-align: center;">18</td> <td style="text-align: center;">[-1.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0...]</td> </tr> </tbody> </table>		<b>outliers_count</b>	<b>outliers_unique</b>	<b>outliers_list</b>	<b>obs</b>	267515	18	[-1.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0...]		
	<b>outliers_count</b>	<b>outliers_unique</b>	<b>outliers_list</b>								
<b>obs</b>	267515	18	[-1.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0...]								

Répartition		
	Count	% valeurs
obs		
-1.0	164	0.0
0.0	1740874	87.0
1.0	44192	2.0
2.0	28984	1.0
3.0	23176	1.0
4.0	23786	1.0
5.0	2925	0.0
6.0	22306	1.0
7.0	4818	0.0
8.0	21345	1.0
9.0	7049	0.0
10.0	2320	0.0
11.0	4577	0.0
12.0	12122	1.0
13.0	34044	2.0
14.0	14764	1.0
15.0	9904	0.0
16.0	10625	1.0
17.0	414	0.0
NaN	1006	0.0
<b>Remarque</b>		
Les valeurs NaN peuvent être remplacées par -1 qui signifie « non renseigné ».		

The figure is a heatmap titled 'obs' showing the distribution of 'obs' values across 18 modalities (0 to 17) over 18 years (2005 to 2022). The y-axis is labeled 'Count' and ranges from 0.00 to 1.75e6. The x-axis is labeled 'Modalités' and ranges from 0 to 17. A color scale on the left indicates the count, with dark purple for 0.00 and yellow for 1.75e6. The legend is titled 'Years' and lists the years from 2005 to 2022. The heatmap shows that most observations are in modalities 0, 1, and 2, with counts reaching up to 1.75e6. Modalities 3 through 17 have much lower counts, generally below 0.5e6.

## f. obsm

Description	Obstacle mobile heurté
Modalités	<ul style="list-style-type: none"> <li>- -1 : Non renseigné</li> <li>- 0 : Aucun</li> <li>- 1 : Piéton</li> <li>- 2 : Véhicule</li> <li>- 4 : Véhicule sur rail</li> <li>- 5 : Animal domestique</li> <li>- 6 : Animal sauvage</li> <li>- 9 : Autre</li> </ul>

Type	[2005-2008 ; 2019-2022] : int64 [2009-2018] : float64																														
Etendue des valeurs	<b>count unique top freq</b> <b>obsm</b> 2008617 8 2.0 1352641																														
Valeurs nulles	<b>Type Val_notnull Val_null %_null</b> <b>obsm</b> float64 2008617 778 0.04																														
Outliers	<b>outliers_count outliers_unique outliers_list</b> <b>obsm</b> 36611 5 [-1.0, 4.0, 5.0, 6.0, 9.0]																														
Répartition	<p><b>Count % valeurs</b></p> <table border="1"> <thead> <tr> <th></th><th>Count</th><th>% valeurs</th></tr> </thead> <tbody> <tr> <td>-1.0</td><td>199</td><td>0.0</td></tr> <tr> <td>0.0</td><td>416695</td><td>21.0</td></tr> <tr> <td>1.0</td><td>202670</td><td>10.0</td></tr> <tr> <td>2.0</td><td>1352641</td><td>67.0</td></tr> <tr> <td>4.0</td><td>1889</td><td>0.0</td></tr> <tr> <td>5.0</td><td>1826</td><td>0.0</td></tr> <tr> <td>6.0</td><td>3840</td><td>0.0</td></tr> <tr> <td>9.0</td><td>28857</td><td>1.0</td></tr> <tr> <td>NaN</td><td>778</td><td>0.0</td></tr> </tbody> </table>		Count	% valeurs	-1.0	199	0.0	0.0	416695	21.0	1.0	202670	10.0	2.0	1352641	67.0	4.0	1889	0.0	5.0	1826	0.0	6.0	3840	0.0	9.0	28857	1.0	NaN	778	0.0
	Count	% valeurs																													
-1.0	199	0.0																													
0.0	416695	21.0																													
1.0	202670	10.0																													
2.0	1352641	67.0																													
4.0	1889	0.0																													
5.0	1826	0.0																													
6.0	3840	0.0																													
9.0	28857	1.0																													
NaN	778	0.0																													
Evolution	<p>Evolution de la distribution obsm</p>																														
Remarque	Les valeurs NaN peuvent être remplacées par -1 qui signifie « non renseigné ».																														

## g. choc

Description	Point de choc initial.																																										
Modalités	<ul style="list-style-type: none"> <li>- -1 : Non renseigné</li> <li>- 0 : Aucun</li> <li>- 1 : Avant</li> <li>- 2 : Avant droit</li> <li>- 3 : Avant gauche</li> <li>- 4 : Arrière</li> <li>- 5 : Arrière droit</li> <li>- 6 : Arrière gauche</li> <li>- 7 : Côté droit</li> <li>- 8 : Côté gauche</li> <li>- 9 : Chocs multiples (tonneaux)</li> </ul>																																										
Type	[2005-2008 ; 2019-2022] : int64 [2009-2018] : float64																																										
Etendue des valeurs	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">count</th><th style="text-align: center;">unique</th><th style="text-align: center;">top</th><th style="text-align: center;">freq</th></tr> </thead> <tbody> <tr> <td>choc</td><td style="text-align: center;">2008998</td><td style="text-align: center;">11</td><td style="text-align: center;">1.0</td><td style="text-align: center;">738510</td></tr> </tbody> </table>		count	unique	top	freq	choc	2008998	11	1.0	738510																																
	count	unique	top	freq																																							
choc	2008998	11	1.0	738510																																							
Valeurs nulles	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">Type</th><th style="text-align: center;">Val_notnull</th><th style="text-align: center;">Val_null</th><th style="text-align: center;">%_null</th></tr> </thead> <tbody> <tr> <td>choc</td><td style="text-align: center;">float64</td><td style="text-align: center;">2008998</td><td style="text-align: center;">397</td><td style="text-align: center;">0.02</td></tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	choc	float64	2008998	397	0.02																																
	Type	Val_notnull	Val_null	%_null																																							
choc	float64	2008998	397	0.02																																							
Outliers	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">outliers_count</th><th style="text-align: center;">outliers_unique</th><th style="text-align: center;">outliers_list</th></tr> </thead> <tbody> <tr> <td>choc</td><td style="text-align: center;">31156</td><td style="text-align: center;">1</td><td style="text-align: center;">[ 9.0 ]</td></tr> </tbody> </table>		outliers_count	outliers_unique	outliers_list	choc	31156	1	[ 9.0 ]																																		
	outliers_count	outliers_unique	outliers_list																																								
choc	31156	1	[ 9.0 ]																																								
Répartition	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">Count</th><th style="text-align: center;">% valeurs</th></tr> </thead> <tbody> <tr> <td>choc</td><td></td><td></td></tr> <tr> <td>-1.0</td><td style="text-align: center;">197</td><td style="text-align: center;">0.0</td></tr> <tr> <td>0.0</td><td style="text-align: center;">133592</td><td style="text-align: center;">7.0</td></tr> <tr> <td>1.0</td><td style="text-align: center;">738510</td><td style="text-align: center;">37.0</td></tr> <tr> <td>2.0</td><td style="text-align: center;">235373</td><td style="text-align: center;">12.0</td></tr> <tr> <td>3.0</td><td style="text-align: center;">289599</td><td style="text-align: center;">14.0</td></tr> <tr> <td>4.0</td><td style="text-align: center;">190557</td><td style="text-align: center;">9.0</td></tr> <tr> <td>5.0</td><td style="text-align: center;">53591</td><td style="text-align: center;">3.0</td></tr> <tr> <td>6.0</td><td style="text-align: center;">67943</td><td style="text-align: center;">3.0</td></tr> <tr> <td>7.0</td><td style="text-align: center;">122529</td><td style="text-align: center;">6.0</td></tr> <tr> <td>8.0</td><td style="text-align: center;">145951</td><td style="text-align: center;">7.0</td></tr> <tr> <td>9.0</td><td style="text-align: center;">31156</td><td style="text-align: center;">2.0</td></tr> <tr> <td>NaN</td><td style="text-align: center;">397</td><td style="text-align: center;">0.0</td></tr> </tbody> </table> <p>The heatmap displays the distribution of shock types over time. The x-axis represents the shock type ('Modalités') from 0 to 9, and the y-axis represents the year ('Years') from 2005 to 2022. The color intensity represents the count of occurrences. The most frequent shock type is 'Avant' (Modalité 1), followed by 'Avant droit' (Modalité 3). Other shock types like 'Arrière' (Modalité 4) and 'Côté' (Modalités 7, 8) are less frequent.</p>		Count	% valeurs	choc			-1.0	197	0.0	0.0	133592	7.0	1.0	738510	37.0	2.0	235373	12.0	3.0	289599	14.0	4.0	190557	9.0	5.0	53591	3.0	6.0	67943	3.0	7.0	122529	6.0	8.0	145951	7.0	9.0	31156	2.0	NaN	397	0.0
	Count	% valeurs																																									
choc																																											
-1.0	197	0.0																																									
0.0	133592	7.0																																									
1.0	738510	37.0																																									
2.0	235373	12.0																																									
3.0	289599	14.0																																									
4.0	190557	9.0																																									
5.0	53591	3.0																																									
6.0	67943	3.0																																									
7.0	122529	6.0																																									
8.0	145951	7.0																																									
9.0	31156	2.0																																									
NaN	397	0.0																																									

<b>Evolution</b>	<p style="text-align: center;">Evolution de la distribution choc</p> <p style="text-align: right;">Modalités</p> <ul style="list-style-type: none"> <li>0</li> <li>1</li> <li>2</li> <li>3</li> <li>4</li> <li>5</li> <li>6</li> <li>7</li> <li>8</li> <li>9</li> <li>-1</li> </ul>
<b>Remarque</b>	Les valeurs NaN peuvent être remplacées par -1 qui signifie « non renseigné ».

## h. manv

<b>Description</b>	Manoeuvre principale avant l'accident.
<b>Modalités</b>	<ul style="list-style-type: none"> <li>- -1 : Non renseigné</li> <li>- 0 : Inconnue</li> <li>- 1 : Sans changement de direction</li> <li>- 2 : Même sens, même file</li> <li>- 3 : Entre 2 files</li> <li>- 4 : En marche arrière</li> <li>- 5 : A contresens</li> <li>- 6 : En franchissant le terre-plein central</li> <li>- 7 : Dans le couloir bus, dans le même sens</li> <li>- 8 : Dans le couloir bus, dans le sens inverse</li> <li>- 9 : En s'insérant</li> <li>- 10 : En faisant demi-tour sur la chaussée</li> <li>- 11 : Changeant de file A gauche</li> <li>- 12 : Changeant de file A droite</li> <li>- 13 : Déporté A gauche</li> <li>- 14 : Déporté A droite</li> <li>- 15 : Tournant A gauche</li> <li>- 16 : Tournant A droite</li> <li>- 17 : Dépassant A gauche</li> <li>- 18 : Dépassant A droite</li> <li>- 19 : Traversant la chaussée</li> <li>- 20 : Manœuvre de stationnement</li> <li>- 21 : Manœuvre d'évitement</li> <li>- 22 : Ouverture de porte</li> <li>- 23 : Arrêté (hors stationnement)</li> <li>- 24 : En stationnement (avec occupants)</li> <li>- 25 : Circulant sur trottoir</li> <li>- 26 : Autres manœuvres</li> </ul>
<b>Type</b>	[2005-2008 ; 2019-2022] : int64 [2009-2018] : float64

Etendue des valeurs	<table border="1"> <thead> <tr> <th></th><th>count</th><th>unique</th><th>top</th><th>freq</th></tr> </thead> <tbody> <tr> <td><b>manv</b></td><td>2008927</td><td>28</td><td>1.0</td><td>863725</td></tr> </tbody> </table>		count	unique	top	freq	<b>manv</b>	2008927	28	1.0	863725																													
	count	unique	top	freq																																				
<b>manv</b>	2008927	28	1.0	863725																																				
Valeurs nulles	<table border="1"> <thead> <tr> <th></th><th>Type</th><th>Val_notnull</th><th>Val_null</th><th>%_null</th></tr> </thead> <tbody> <tr> <td><b>manv</b></td><td>float64</td><td>2008927</td><td>468</td><td>0.02</td></tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>manv</b>	float64	2008927	468	0.02																													
	Type	Val_notnull	Val_null	%_null																																				
<b>manv</b>	float64	2008927	468	0.02																																				
Outliers	<table border="1"> <thead> <tr> <th></th><th>outliers_count</th><th>outliers_unique</th><th>outliers_list</th></tr> </thead> <tbody> <tr> <td><b>manv</b></td><td>0</td><td>0</td><td>[ ]</td></tr> </tbody> </table> <p style="text-align: center;">Boxplots pour: manv</p>		outliers_count	outliers_unique	outliers_list	<b>manv</b>	0	0	[ ]																															
	outliers_count	outliers_unique	outliers_list																																					
<b>manv</b>	0	0	[ ]																																					
<b>Répartition</b> <table border="1"> <thead> <tr> <th></th><th>Count</th><th>% valeurs</th></tr> </thead> <tbody> <tr> <td><b>manv</b></td><td></td><td></td></tr> <tr> <td><b>-1.0</b></td><td>142</td><td>0.01</td></tr> <tr> <td><b>0.0</b></td><td>156498</td><td>7.79</td></tr> <tr> <td><b>1.0</b></td><td>863725</td><td>42.98</td></tr> <tr> <td><b>2.0</b></td><td>233007</td><td>11.6</td></tr> <tr> <td><b>3.0</b></td><td>15728</td><td>0.78</td></tr> <tr> <td>...</td><td>...</td><td>...</td></tr> <tr> <td><b>23.0</b></td><td>52994</td><td>2.64</td></tr> <tr> <td><b>24.0</b></td><td>7274</td><td>0.36</td></tr> <tr> <td><b>25.0</b></td><td>1040</td><td>0.05</td></tr> <tr> <td><b>26.0</b></td><td>12678</td><td>0.63</td></tr> <tr> <td><b>Nan</b></td><td>468</td><td>0.02</td></tr> </tbody> </table> <p>29 rows x 2 columns</p>		Count	% valeurs	<b>manv</b>			<b>-1.0</b>	142	0.01	<b>0.0</b>	156498	7.79	<b>1.0</b>	863725	42.98	<b>2.0</b>	233007	11.6	<b>3.0</b>	15728	0.78	...	...	...	<b>23.0</b>	52994	2.64	<b>24.0</b>	7274	0.36	<b>25.0</b>	1040	0.05	<b>26.0</b>	12678	0.63	<b>Nan</b>	468	0.02	<p>Years</p> <ul style="list-style-type: none"> <li>2005</li> <li>2006</li> <li>2007</li> <li>2008</li> <li>2009</li> <li>2010</li> <li>2011</li> <li>2012</li> <li>2013</li> <li>2014</li> <li>2015</li> <li>2016</li> <li>2017</li> <li>2018</li> <li>2019</li> <li>2020</li> <li>2021</li> <li>2022</li> </ul>
	Count	% valeurs																																						
<b>manv</b>																																								
<b>-1.0</b>	142	0.01																																						
<b>0.0</b>	156498	7.79																																						
<b>1.0</b>	863725	42.98																																						
<b>2.0</b>	233007	11.6																																						
<b>3.0</b>	15728	0.78																																						
...	...	...																																						
<b>23.0</b>	52994	2.64																																						
<b>24.0</b>	7274	0.36																																						
<b>25.0</b>	1040	0.05																																						
<b>26.0</b>	12678	0.63																																						
<b>Nan</b>	468	0.02																																						

<b>Remarque</b>	Les valeurs NaN peuvent être remplacées par -1 qui signifie « non renseigné ».
-----------------	--

### i. num\_veh

<b>Description</b>	Identifiant du véhicule repris pour chacun des usagers occupant ce véhicule (y compris les piétons qui sont rattachés aux véhicules qui les ont heurtés) - Code alphanumérique.																																											
<b>Type</b>	object																																											
<b>Etendue des valeurs</b>	<table> <thead> <tr> <th></th> <th>count</th> <th>unique</th> <th>top</th> <th>freq</th> </tr> </thead> <tbody> <tr> <td><b>num_veh</b></td> <td>2009395</td> <td>189</td> <td>A01</td> <td>1160074</td> </tr> </tbody> </table>						count	unique	top	freq	<b>num_veh</b>	2009395	189	A01	1160074																													
	count	unique	top	freq																																								
<b>num_veh</b>	2009395	189	A01	1160074																																								
<b>Valeurs nulles</b>	<table> <thead> <tr> <th></th> <th>Type</th> <th>Val_notnull</th> <th>Val_null</th> <th>%_null</th> </tr> </thead> <tbody> <tr> <td><b>num_veh</b></td> <td>object</td> <td>2009395</td> <td>0</td> <td>0.0</td> </tr> </tbody> </table>						Type	Val_notnull	Val_null	%_null	<b>num_veh</b>	object	2009395	0	0.0																													
	Type	Val_notnull	Val_null	%_null																																								
<b>num_veh</b>	object	2009395	0	0.0																																								
<b>Outliers</b>	<table> <thead> <tr> <th></th> <th>outliers_count</th> <th>outliers_unique</th> <th colspan="2"><b>outliers_list</b></th> </tr> </thead> <tbody> <tr> <td><b>num_veh</b></td> <td>40101</td> <td>184</td> <td colspan="2" rowspan="2">[ A02, A03, A04, A05, A06, A07, A08, A09, A27,...</td> </tr> </tbody> </table>						outliers_count	outliers_unique	<b>outliers_list</b>		<b>num_veh</b>	40101	184	[ A02, A03, A04, A05, A06, A07, A08, A09, A27,...																														
	outliers_count	outliers_unique	<b>outliers_list</b>																																									
<b>num_veh</b>	40101	184	[ A02, A03, A04, A05, A06, A07, A08, A09, A27,...																																									
<b>Répartition</b>	<table> <thead> <tr> <th></th> <th>Count</th> <th>% valeurs</th> </tr> </thead> <tbody> <tr> <td><b>num_veh</b></td> <td></td> <td></td> </tr> <tr> <td>A01</td> <td>1160074</td> <td>58.0</td> </tr> <tr> <td>A02</td> <td>527</td> <td>0.0</td> </tr> <tr> <td>A03</td> <td>33</td> <td>0.0</td> </tr> <tr> <td>A04</td> <td>7</td> <td>0.0</td> </tr> <tr> <td>A05</td> <td>6</td> <td>0.0</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>ZB01</td> <td>1</td> <td>0.0</td> </tr> <tr> <td>ZZ01</td> <td>8</td> <td>0.0</td> </tr> <tr> <td>[01</td> <td>20</td> <td>0.0</td> </tr> <tr> <td>\01</td> <td>3</td> <td>0.0</td> </tr> <tr> <td>]01</td> <td>1</td> <td>0.0</td> </tr> </tbody> </table> <p>189 rows × 2 columns</p>						Count	% valeurs	<b>num_veh</b>			A01	1160074	58.0	A02	527	0.0	A03	33	0.0	A04	7	0.0	A05	6	0.0	...	...	...	ZB01	1	0.0	ZZ01	8	0.0	[01	20	0.0	\01	3	0.0	]01	1	0.0
	Count	% valeurs																																										
<b>num_veh</b>																																												
A01	1160074	58.0																																										
A02	527	0.0																																										
A03	33	0.0																																										
A04	7	0.0																																										
A05	6	0.0																																										
...	...	...																																										
ZB01	1	0.0																																										
ZZ01	8	0.0																																										
[01	20	0.0																																										
\01	3	0.0																																										
]01	1	0.0																																										

### j. id\_vehicule

<b>Description</b>	Identifiant unique du véhicule repris pour chacun des usagers occupant ce véhicule - code numérique.
<b>Type</b>	object

<b>Etendue des valeurs</b>		<b>count</b>	<b>unique</b>	<b>top</b>	<b>freq</b>
	<b>id_vehicule</b>	373584	373584	100 882	1
<b>Valeurs nulles</b>	<b>Type</b>	<b>Val_notnull</b>	<b>Val_null</b>	<b>%_null</b>	
	<b>num_veh</b>	object	2009395	0	0.0
<b>Valeurs uniques</b>	<b>Type</b>	<b>Val_notnull</b>	<b>Val_null</b>	<b>%_null</b>	
	<b>id_vehicule</b>	object	373584	1635811	81.41
<b>Outliers</b>	<b>outliers_count</b>	<b>outliers_unique</b>			<b>outliers_list</b>
	<b>id_vehicule</b>	373584	373584	[ 100 882, 100 883, 100 884, 100 885, 100 886,...	
<b>Répartition</b>	<b>Count</b>	<b>% valeurs</b>			
	<b>id_vehicule</b>				
	<b>100 882</b>	1	0.0		
	<b>100 883</b>	1	0.0		
	<b>100 884</b>	1	0.0		
	<b>100 885</b>	1	0.0		
	<b>100 886</b>	1	0.0		
	<b>...</b>	...	...		
	<b>813 950</b>	1	0.0		
	<b>813 951</b>	1	0.0		
	<b>813 952</b>	1	0.0		
	<b>813 953</b>	1	0.0		
	<b>Nan</b>	1635811	81.0		
	373585 rows × 2 columns				
<b>Remarque</b>	Apparaît à partir de 2019.				

## k. motor

<b>Description</b>	Type de motorisation du véhicule.
<b>Modalités</b>	<ul style="list-style-type: none"> <li>- -1 : Non renseigné</li> <li>- 0 : Inconnue</li> <li>- 1 : Hydrocarbures</li> <li>- 2 : Hybride électrique</li> <li>- 3 : Electrique</li> </ul>

	<ul style="list-style-type: none"> <li>- 4 : Hydrogène</li> <li>- 5 : Humaine</li> <li>- 6 : Autre</li> </ul>																																	
Type	[2019-2022] : int64																																	
Etendue des valeurs	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">count</th><th style="text-align: center;">unique</th><th style="text-align: center;">top</th><th style="text-align: center;">freq</th></tr> </thead> <tbody> <tr> <td><b>motor</b></td><td style="text-align: center;">373584</td><td style="text-align: center;">8</td><td style="text-align: center;">1.0</td><td style="text-align: center;">304898</td></tr> </tbody> </table>		count	unique	top	freq	<b>motor</b>	373584	8	1.0	304898																							
	count	unique	top	freq																														
<b>motor</b>	373584	8	1.0	304898																														
Valeurs nulles	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">Type</th><th style="text-align: center;">Val_notnull</th><th style="text-align: center;">Val_null</th><th style="text-align: center;">%_null</th></tr> </thead> <tbody> <tr> <td><b>motor</b></td><td style="text-align: center;">float64</td><td style="text-align: center;">373584</td><td style="text-align: center;">1635811</td><td style="text-align: center;">81.41</td></tr> </tbody> </table>		Type	Val_notnull	Val_null	%_null	<b>motor</b>	float64	373584	1635811	81.41																							
	Type	Val_notnull	Val_null	%_null																														
<b>motor</b>	float64	373584	1635811	81.41																														
Outliers	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">outliers_count</th><th style="text-align: center;">outliers_unique</th><th style="text-align: right; vertical-align: bottom;"><b>outliers_list</b></th></tr> </thead> <tbody> <tr> <td><b>motor</b></td><td style="text-align: center;">68686</td><td style="text-align: center;">7</td><td style="text-align: right; vertical-align: bottom;">[ -1.0, 0.0, 2.0, 3.0, 4.0, 5.0, 6.0 ]</td></tr> </tbody> </table>		outliers_count	outliers_unique	<b>outliers_list</b>	<b>motor</b>	68686	7	[ -1.0, 0.0, 2.0, 3.0, 4.0, 5.0, 6.0 ]																									
	outliers_count	outliers_unique	<b>outliers_list</b>																															
<b>motor</b>	68686	7	[ -1.0, 0.0, 2.0, 3.0, 4.0, 5.0, 6.0 ]																															
Répartition	<p><b>Count % valeurs</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">Count</th><th style="text-align: center;">% valeurs</th></tr> </thead> <tbody> <tr> <td><b>motor</b></td><td></td><td></td></tr> <tr> <td><b>-1.0</b></td><td style="text-align: center;">865</td><td style="text-align: center;">0.0</td></tr> <tr> <td><b>0.0</b></td><td style="text-align: center;">28200</td><td style="text-align: center;">1.0</td></tr> <tr> <td><b>1.0</b></td><td style="text-align: center;">304898</td><td style="text-align: center;">15.0</td></tr> <tr> <td><b>2.0</b></td><td style="text-align: center;">5558</td><td style="text-align: center;">0.0</td></tr> <tr> <td><b>3.0</b></td><td style="text-align: center;">10997</td><td style="text-align: center;">1.0</td></tr> <tr> <td><b>4.0</b></td><td style="text-align: center;">191</td><td style="text-align: center;">0.0</td></tr> <tr> <td><b>5.0</b></td><td style="text-align: center;">19685</td><td style="text-align: center;">1.0</td></tr> <tr> <td><b>6.0</b></td><td style="text-align: center;">3190</td><td style="text-align: center;">0.0</td></tr> <tr> <td><b>Nan</b></td><td style="text-align: center;">1635811</td><td style="text-align: center;">81.0</td></tr> </tbody> </table>		Count	% valeurs	<b>motor</b>			<b>-1.0</b>	865	0.0	<b>0.0</b>	28200	1.0	<b>1.0</b>	304898	15.0	<b>2.0</b>	5558	0.0	<b>3.0</b>	10997	1.0	<b>4.0</b>	191	0.0	<b>5.0</b>	19685	1.0	<b>6.0</b>	3190	0.0	<b>Nan</b>	1635811	81.0
	Count	% valeurs																																
<b>motor</b>																																		
<b>-1.0</b>	865	0.0																																
<b>0.0</b>	28200	1.0																																
<b>1.0</b>	304898	15.0																																
<b>2.0</b>	5558	0.0																																
<b>3.0</b>	10997	1.0																																
<b>4.0</b>	191	0.0																																
<b>5.0</b>	19685	1.0																																
<b>6.0</b>	3190	0.0																																
<b>Nan</b>	1635811	81.0																																
Evolution	<p><b>Evolution de la distribution motor</b></p>																																	
Remarque	Les valeurs Nan peuvent être remplacées par -1 qui signifie « non »																																	

	renseigné ».
--	--------------