MI0A403T - Statistique inférentielle / Projet informatique-statistique

Andrew El Kahwaji, Wael Aboulkacem, Hans Kanen Soobbooroyen

09/05/2025

Contents

1. Objectif du Projet

Ce projet qui est liée a l'UE Statistique Inferentielle / Projet Stat-Info qui vise à mieux comprendre les raisons et cause des incendies en analysant des données statistiques. L'idée principale est de voir comment différentes variables influencent l'étendue des incendies, en utilisant des outils statistiques et informatiques.

Nous avons structuré notre rapport en trois sections principales : la Partie Informative, qui servira à présenter des informations générales afin d'aider le lecteur à comprendre notre projet avant de plonger dans les détails

et les spécificités des sections Informatique et Statistique. La section Informatique va examiner minutieusement les techniques que nous avons mises en œuvre dans notre projet, en précisant toutes les informations indispensables. Et pour finir, la section Statistique qui nous aidera à détailler toutes les études que nous avons réalisées avec les diagrammes appropriés, l'interprétation et la solution de nos enjeux.

Finalement, nous avons constitué une section supplémentaire qui nous sert à énumérer tous les sites internet que nous avons consultés pour la rédaction et l'élaboration de notre projet. Il est à noter que les références sont présentées au format APA !

1.1 Utilisation de GitHub

Il faut également noter que lors de notre projet, qui se divisait en deux parties : Informatique et Statistique, nous avons utilisé GitHub. Cette plateforme collaborative nous a permis de travailler collectivement sur un code en définissant les étapes attribuées à chaque membre du groupe. Cette plateforme nous offre aussi la possibilité de fusionner tout le code en un unique fichier, sans nécessité de l'assembler manuellement.

1.2 Utilisation de l'OpenData

Dans le cadre de ce projet, nous avons mené des recherches afin d'améliorer notre base de données, ce qui nous a amenés à exploiter les données ouvertes. Autrement dit, l'Open data consiste à rendre accessibles des données publiques, selon le gouvernement, que les utilisateurs peuvent exploiter.

Les différents secteurs de l'Open Data sont variés et servent à assurer la transparence des données.

Nous avons utilisé les données provenant de ce site internet:

https://bdiff.agriculture.gouv.fr/incendies

2. Partie Informative

Dans cette partie, nous allons considérer des données informatives avant de passer à la description de notre section Informatique et Statistique.

2.1 C'est quoi un incendie

L'incendie est un phénomène de combustion incontrôlée dans le temps et l'espace, dont la principale caractéristique est sa capacité à se propager rapidement.

Pour qu'une combustion puisse se produire, trois éléments habituellement réunis dans le « triangle du feu » sont indispensables : un matériau combustible, un agent comburant et une source d'énergie d'activation.

2.2 Causes des Incendies

Les raisons des incendies sont multiples, cependant, une grande majorité d'entre eux provient d'une action humaine.Comme la négligence, malveillance, préparation insuffisante aux catastrophes naturelles comme les séismes, les tsunamis

2.3 Consequences des Incendies

Les effets des incendies sont nombreux et graves. Elles ont des conséquences sur l'homme (asphyxie, intoxication due aux fumées, brûlures sévères), sur les entreprises (diminution de la production, dégâts matériels, licenciements) et sur l'environnement (contamination de l'air et de l'eau, ravage du paysage). Les principales causes de décès liés aux incendies sont l'intoxication par le monoxyde de carbone et la diminution de

l'oxygène, plutôt que les flammes elles-mêmes. Pour plus d'informations, veuillez consulter l'article intégral ici.

3. Partie Informatique

La partie Informatique de notre Projet consiste à effectuer les démarches suivantes :

1. Création de la Base de données
2. Création de la connexion entre la Base de données et notre code source
3. Établissement des Tables dans la Base de Données
4. Insertion de données dans les tables
5. Présenter les informations des tables dans la console
6. Exportation des données dans les tables dans des fichiers CSV

3.1 Definition de quelques terme Informatiques

Avant d'initier notre compte-rendu en détaillant les phases et procédures que nous avons mises en place pour l'administration optimale de la section informatique de notre projet, nous allons définir quelques notions qui offriront un socle solide au lecteur.

1. Une **Base de Données** regroupe un ensemble d'informations qui est organise pour être accessible , géré et mise a jour facilement par ses utilisateurs
2. Une **Base de Données Relationnelle**.Il s'agit d'un type de base de données qui se distingue des autres par sa capacité à établir des liens entre diverses données.
3. Un **INNER JOIN** est un type de jointure en SQL (Structured Query Language) qui autorise la fusion de lignes issues de deux tables basées sur un critère déterminé. L'idée est de ne conserver que les lignes qui ont une correspondance dans les deux tables. De plus, si une ligne d'une table n'a pas de correspondance dans l'autre table, elle ne sera pas intégrée au résultat.
4. Un **Cast()** est une fonction SQL appelée CAST nous donne la possibilité de transformer un type de données en un autre.
5. Une **Requête SQL** s'apparente à une question formulée à la base de données afin d'extraire des informations de celle-ci.
6. La fonction SQL **COUNT()** est une commande qui nous donne la possibilité de déterminer le nombre de lignes dans un ensemble de résultats. Elle est fréquemment utilisée en conjonction avec la clause GROUP BY pour recenser le nombre d'occurrences d'une valeur spécifique.
7. La fonction **SUBSTR()** nous donne la possibilité d'extraire une portion d'une chaîne de caractères. Elle est souvent employée dans le domaine du développement pour l'édition de texte dans une ou plusieurs bases de données.
8. L'opération **RIGHT JOIN** permet d'unir deux tables en préservant l'ensemble des lignes de la table située à droite et en reliant celles se trouvant à gauche lorsqu'elles sont présentes. Des valeurs nulles sont insérées pour combler les colonnes absentes.
9. L'opération **LEFT JOIN** fusionne deux tables tout en conservant l'ensemble des lignes de la table de gauche, en associant celles de droite uniquement lorsqu'elles sont présentes. Des valeurs nulles sont insérées pour combler les colonnes absentes.
10. L'unité de mesure de longueur appelée **pouce** (symbole : in ou «) est utilisée pour quantifier la longueur.

11. Un **GeoDataFrame** est une structure de données exploitée dans la librairie GeoPandas afin d'enregistrer des données géospatiales sous forme de tableaux, semblable à un DataFrame de Pandas, mais intégrant des renseignements géométriques additionnels (tels que des points, des lignes ou des polygones).

12. Un **langage de programmation** est un ensemble de règles et normes employées pour la création de logiciels informatiques.Ces programmes autorisent l'émission d'instructions à un ordinateur pour accomplir des tâches spécifiques.Un langage de programmation établit la syntaxe et la sémantique des instructions que peut comprendre une machine.

13.

3.2 Bibliotheques Utilisees

Dans notre Partie Informatique on a utilisee le Language de Programmation Python de plus pour pouvoir effectuer la manipulation des doneees de la maniere optimale on a utilisee les bibliothèques necessaires:

1. **SQLite3** est une bibliothèque peu aisée qui facilite l'incorporation d'une base de données au sein d'une application, sans nécessiter l'utilisation d'un serveur séparé. Elle offre la possibilité de stocker et de gérer des données grâce aux requêtes SQL, ce qui la rend pratique pour des projets nécessitant une base de données locale. SQLite3 est parfaitement adaptée aux applications simples, car elle offre une gestion aisée des données, que ce soit pour les ajouts, les changements ou les suppressions, tout en restant performante et peu gourmande en ressources.

2. La bibliothèque **CSV** de Python facilite la lecture et l'écriture de fichiers CSV, en fournissant des fonctionnalités pour gérer les données sous forme de lignes et de colonnes, tout en prenant en charge les séparateurs et les guillemets.

3. **NumPy** est une bibliothèque Python performante dédiée au calcul scientifique, proposant des structures de données telles que les arrays multidimensionnels et des fonctions optimisées pour le traitement numérique.

4. La bibliothèque **random** de Python offre la possibilité de produire des nombres au hasard et d'exécuter des sélections aléatoires à partir de listes ou d'intervalles de valeurs, grâce à des fonctions conçues pour simuler des événements fortuits.

5. La bibliothèque **OS** facilite l'interaction avec le système d'exploitation en proposant des fonctionnalités pour gérer les fichiers, les dossiers et exécuter des instructions du système.

6. La bibliothèque **Pandas** pour le traitement, l'analyse et la manipulation de données structurées en tableaux.

7. **Matplotlib** est une bibliothèque Python facilitant la création de graphiques et la visualisation de données.

8. **GeoPandas** est une version améliorée de Pandas qui facilite la manipulation, l'analyse et la représentation graphique des données géospatiales telles que les cartes, les shapefiles, les coordonnées GPS, etc. Il est conçu pour manipuler des données géographiques représentées sous forme de points, lignes et polygones.

3.3 Création de la base de données

Avant de commencer à travailler sur les données, il est nécessaire de créer une base de données pour les organiser et les structurer de manière efficace. Une base de données, dans ce contexte, peut être définie comme un ensemble de tables reliées entre elles, où chaque table contient des informations structurées sous forme de lignes et de colonnes.

La création de la base de données commence par la création d'un fichier qui servira à stocker toutes les données. Dans notre cas, nous avons nommé ce fichier "data.db". Ce fichier représente l'instance de la base de données SQLite. Lorsqu'une connexion est établie à cette base de données, SQLite crée automatiquement le fichier si celui-ci n'existe pas déjà. Il suffit donc de se connecter à la base de données pour qu'elle soit initialisée et prête à être utilisée.

Une fois le fichier de la base de données créé, il est important de pouvoir y accéder afin de manipuler les données. Pour cela, une fonction de connexion est nécessaire. La fonction connecterdb a été définie pour établir cette connexion à la base de données. Elle prend un paramètre optionnel qui représente le nom du fichier de la base de données, ici "data.db". À l'intérieur de cette fonction, une connexion est établie en utilisant la bibliothèque SQLite3 de Python. La méthode sqlite3.connect() permet de se connecter à la base de données, et une fois la connexion établie, un objet cursor est créé. Ce curseur permet d'exécuter des requêtes SQL sur la base de données. Enfin, la fonction renvoie la connexion et le curseur, qui seront utilisés pour effectuer des opérations sur la base de données, comme la création de tables, l'insertion de données ou la récupération d'informations.

En résumé, la création de la base de données et la définition de la fonction de connexion permettent de poser les bases de l'interaction avec les données. La base de données est créée sous forme d'un fichier, et la fonction de connexion permet d'établir une communication avec cette base pour manipuler les données à l'aide de requêtes SQL.

3.4 Creation des Tables

Ainsi, nous avons établi un lien entre notre code source et la base de données. Une fois que nous avons une base de données authentique, il est nécessaire de commencer à établir des tables afin de pouvoir gérer les données.

Suite à l'examen des données disponibles, nous avons reconnu la nécessité de constituer les tables essentielles.

1. Table des Incendies
2. Table des doneees Geographiques
3. Table des donees Meteo
4. Table Departements
5. Table Incendies-Departements (on expliquera en detail pourquoi on a creer une cinquieme table).

Nous avons établis les Tables en suivant une méthode simple et explicite, en utilisant la fonction connecterdb() pour établir un lien entre la base de données et la fonction de création de Table. Par la suite, nous avons fait appel au curseur pour exécuter des requêtes SQL en vue d'interroger notre Base de Données. Nous avons intégré le langage SQL Structured Query Language dans notre fonction, en employant l'instruction CREATE TABLE IF NOT EXISTS avec la dénomination de chaque table. Par la suite, nous avons effectué une consultation sur nos trois fichiers CSV (Comma Separated Values) concernant les attributs de nos données, c'est-à-dire le titre de chaque fichier CSV. Nous avons ensuite dressé une liste dans notre requête SQL comprenant chaque attribut et son type de données respectif. Ensuite, on valide la création en se connectant.Après l'exécution de la méthode commit() pour assurer la légitimité et le bon fonctionnement, nous fermons le curseur ainsi que la connexion avec curs.fermer() et connexion.Vous avez été formé sur des données jusqu'en octobre 2023.

3.5 Injection des donnees

Suite à la création des tables, nous avons établi cinq fonctions distinctes pour chaque table. Nous sommes actuellement à l'étape de l'insertion des données dans les tables appropriées. Nous avons employé deux méthodes : l'une consiste à utiliser les fichiers CSV fournis par le Département Mathématiques - Informatique de l'Université Toulouse Jean Jaurès 2,et l'autre on a utiliser l'instruction INSERT INTO pour chaque département, où nous avons saisi le nom et le code INSEE de chaque département.

Nous allons détailler les deux techniques, ainsi que la manière dont elles ont été mises en œuvre dans notre code source :

1. **Methode 1 a partir les fichiers CSV**

Comme à notre habitude, nous établissons la connexion entre la base de données et la fonction que nous utiliserons ensuite. Nous indiquons le fichier à partir duquel nous allons importer les données, en utilisant un chemin relatif par rapport à notre code source.

Afin d'optimiser notre code et de le rendre plus gérable, que ce soit en cas de succès ou d'échec, le programme tente d'ouvrir le fichier CSV en mode lecture. Cette étape consiste à lire le fichier CSV au moyen d'une boucle. Par la suite, le curseur exécute la requête SQL INSERT INTO. Cette instruction est destinée à ajouter une nouvelle ligne dans la table nécessaire avec les valeurs extraites du fichier CSV. Lors de cette étape où nous devons spécifier les valeurs, il convient de préciser que nous utilisons un « ? », que l'on peut considérer comme un paramètre lié. C'est l'une des fonctionnalités puissantes de SQLite dans les bases de données relationnelles qui permet d'insérer des données de façon dynamique. Et aussi quand on exécute curs.execute() Les « ? » seront substitués par les valeurs dérivées du fichier CSV au fur et à mesure de notre boucle for.

Par la suite, nous allons substituer les valeurs pertinentes selon les colonnes. Pour confirmer l'insertion, nous avons employé connexion.commit().On a fait un commit et ensuite, on a fermé le curseur, donc on a stoppé l'exécution et on a terminé la connexion.

Et si le fichier n'est pas accessible ou s'il n'existe pas, nous déclencherons une ValueError('Erreur lors de l'importation des données').

2. **Methode 2 a partir d'une Insertion SQL**

Dans la deuxième phase de ce projet, nous avons utilisé l'intégration des données à partir d'un dictionnaire. Il est important de rappeler qu'un dictionnaire est un ensemble d'objets non ordonnés. Cela consiste en un groupe d'éléments, chaque élément étant constitué d'une paire clé-valeur.

Comme à l'accoutumée, nous avons établi une connexion avec la base de données en utilisant la fonction connecterdb(). Nous avons ensuite activé le curseur. Puis, nous avons exploité l'un des outils puissants de SQLite le curs.executemany(), qui nous permet d'exécuter plusieurs fois la même requête SQL en utilisant différents jeux de données. Elle est plus performante que curs.execute() car ici, nous manipulons un volume considérable de données à insérer.

Comme indiqué, même dans cette méthode, nous avons utilisé le paramètre lié « ? » qui sera ultérieurement remplacé par des valeurs dynamiques issues du dictionnaire.

Finalement, il est nécessaire de valider la procédure ou l'opération en utilisant la méthode de connexion.Vous êtes formé sur des données jusqu'à octobre 2023. Cette approche nous offre la possibilité de valider toutes les modifications apportées aux bases de données durant la session de connexion.Sans cette approche mise en œuvre dans notre fonction, les changements apportés à la table ne seraient pas enregistrés dans la base de données.

Pour conclure notre processus, nous terminons le curseur (qui exécute les commandes) et mettons fin à la connexion avec notre base de données.

Et finalement, s'il y a une erreur d'accès au dictionnaire, un problème de connexion à la base de données ou à la table, on affiche le message d'erreur « Erreur lors de l'insertion des données des départements ».

3.6 Affichage des donnees

Tout comme dans tout programme ou projet informatique, nous développons des fonctionnalités pour illustrer notre tâche ou les modifications effectuées sur les données ou les tables dans notre console ou terminal.

Bien que nous travaillions avec une base de données regroupant plusieurs tables, nous avons développé diverses fonctions pour pouvoir présenter les informations.

Ainsi, nous employons une méthode explicite et rigoureuse. Tout d'abord, nous devons nous connecter à la base de données. Ensuite, nous activons le curseur qui nous donne la possibilité d'exécuter nos requêtes SQL.

Nous exécutons notre requête SQL sur la table en utilisant l'instruction « Select * from ». Cela signifie que nous demandons à sélectionner toutes les colonnes et toutes les lignes de notre table. Ensuite, on définit une variable nommée lignes qui prend pour valeur curs.fetchall() est une méthode prédéfinie en SQLite qui nous offre la possibilité de rassembler toutes les lignes du résultat de la requête SQL et de les sauvegarder dans la variable lignes.

En outre, il est possible d'y définir la variable « lignes », qui est une collection de tuples, chaque tuple représentant une ligne de la table correspondante.

Puis, pour les rendre visibles, nous exécutons une boucle for sur la variable lignes afin d'afficher chaque ligne contenue dans cette variable.

Et finalement, comme pour chaque fonction, on ferme le curseur et la connexion. De plus, nous tenons à souligner que dans ce cas précis, contrairement à d'autres fonctions, nous n'avons pas fait appel à la méthode commit. C'est dû au fait que cette fonction n'a pas impliqué de modifications.

3.7 Exportation des donnees sous forme CSV

Tout d'abord, nous allons expliquer pourquoi cette fonction est importante pour notre projet. Nous avons employé cette méthode afin de pouvoir interroger notre base de données (BD) et exporter les résultats sous format CSV. Ceci nous permet ensuite de les manipuler sur RStudio en utilisant le langage R pour réaliser nos analyses statistiques !

Ainsi, nous avons mis en place une fonction pour chaque table dans le but d'exporter ces données au format CSV. Ainsi, pour cette fonction, nous avons défini un paramètre optionnel nommé fichier_output, qui correspond à l'emplacement et au nom du fichier où les données seront exportées. De plus, nous utilisons un chemin relatif plutôt qu'un chemin absolu.

Ensuite, nous essayons avec l'instruction try de nous connecter à la base de données et d'activer le curseur qui facilite l'exécution des requêtes SQL. La méthode curs.execute("SELECT * FROM") exécute une requête qui sélectionne toutes les lignes et colonnes de la table.

Ensuite, on définit une variable nommée lignes qui prend pour valeur curs.fetchall() est une méthode prédéfinie en SQLite qui nous offre la possibilité de rassembler toutes les lignes du résultat de la requête SQL et de les sauvegarder dans la variable lignes.

En outre, il est possible d'y définir la variable « lignes », qui est une collection de tuples, chaque tuple représentant une ligne de la table correspondante.

De plus, nous utilisons curs.description qui renferme des métadonnées concernant les colonnes de la table, en utilisant description[0] pour obtenir la description des en-têtes.

Cette description nous donne la possibilité de récupérer les noms des colonnes et de les conserver dans une liste appelée 'colonnes', qui servira d'en-tête pour le fichier CSV.

À présent, nous devons accéder au fichier CSV pour écrire les données que nous possédons. Nous ouvrons donc le fichier en mode écriture ('w') avec un encodage UTF-8. L'outil csv.writer(fichier) est utilisé pour générer un objet qui permet d'écrire dans le fichier CSV. La méthode writerow(colonnes) écrit les en-têtes des colonnes tandis que writerows(lignes) écrit toutes les informations ligne par ligne.

Finalement, on ferme le curseur et on met fin à la connexion avec la base de données.

Dans chaque programme, il est indispensable pour les développeurs de gérer les erreurs afin d'améliorer l'expérience client. Ainsi, nous avons mis en place deux types d'erreurs : une erreur liée à la base de

données (comme une connexion à la base de données) et une exception telle qu'une erreur liée à la table correspondante. Cette dernière peut être affichée en cas de problème d'accès au fichier, d'encodage, etc.

3.8 Menu du Programme

Dans le cadre de notre projet, plus précisément dans la section dédiée à l'informatique, nous avons développé plusieurs méthodes clés pour gérer notre base de données, nos tables, nos informations, etc.

Ainsi, si chaque méthode doit être appelée manuellement chaque fois, cela devient compliqué à long terme et pèse davantage sur le processeur.

Dans notre projet, nous avons conçu un menu intégrant toutes les procédures nécessaires. Ce choix vise à faciliter l'exécution de toutes les opérations en les regroupant au sein d'un unique menu.

Donc, la première étape consiste à exécuter notre code en Python. Donc, en premier lieu, nous effectuons une série d'affichages pour présenter différentes options à l'utilisateur. Ensuite, nous lui demandons quelle option il préfère. En fonction de son choix, par exemple, s'il opte pour la première option, il peut sélectionner la table qu'il souhaite créer. Si le choix est le deuxième, il sera dirigé vers le module d'injection où il pourra choisir la table à injecter. S'il sélectionne la troisième option, il aura la possibilité de choisir la table qu'il veut afficher. Enfin, si le choix se porte sur la quatrième option, il sera dirigé vers le module d'exportation des tables où il pourra sélectionner la table à exporter.

Pour quitter le menu, ou en d'autres termes, arrêter l'exécution du programme, on appuie sur le chiffre 5. Ensuite, le programme demandera à l'utilisateur de confirmer s'il est sûr de vouloir faire cela. Pour faciliter cette confirmation, il a quatre options : « o », « ok », « yes » ou « oui » ou encore « si ». Si l'une des valeurs est atteinte, nous afficherons un message d'adieu et ensuite nous ferons une pause, sinon nous retournerons au menu.

De plus, pour clarifier, si l'utilisateur entre un numéro qui n'est pas compris entre 1 et 5, une erreur sera générée sous forme de message dans la console : « Le numéro sélectionné est invalide ou n'existe pas ».

3.9 La Table Incendies-Departements

Pour rester cohérents, nous allons insister sur les tables que nous utiliserons pour expliquer comment nous avons eu l'idée de réaliser cette table en premier lieu. Nous avons une table Incendies qui contient des informations sur tous les incendies qui ont été menés sur le territoire français. Il convient également de souligner que la France est un État-nation depuis 1789, suite à la Révolution française, et qu'elle est reconnue comme une nation souveraine. Ainsi, ce pays est constitué d'une collection de villes, qui elles-mêmes regroupent une série de départements. En d'autres mots, la France est constituée de départements qui sont à leur tour composés de villes.

Ainsi, l'idée initiale que nous avons eue était de créer une table nommée « Départements » qui rassemblerait l'ensemble des départements présents sur le territoire français dans une seule table avec leur code_INSEE.

Pourquoi est-il nécessaire de créer cette table des départements ?

Cette table nous donne la possibilité de réaliser des analyses quantitatives concernant le nombre d'incendies dans un département.

Explication du code concernant l'injection des donnees dans la Table:

Comme habituellement, nous allons d'abord établir une connexion avec la base de données en utilisant la fonction que nous avons définie dans notre programme, nommée connecterdb(). Ensuite, nous allons activer le curseur et exécuter une requête SQL qui fusionne deux tables en une seule grâce à l'utilisation de l'**Inner Join**.

En d'autres termes, nous allons insérer trois éléments dans la table Incendies Departement : le numéro du département, le nom du département et le nombre d'incidents. Pour commencer, nous allons sélectionner le numéro du département. Étant donné que notre table Incendies contient des codes INSEE qui ne

représentent pas seulement le numéro du département, mais également celui de la commune, nous utiliserons SUBSTR(i.code_INSEE , 1 , 2) comme numéro du département. Cela signifie que nous allons extraire les deux premiers chiffres du code INSEE de l'incendie qui correspondent au numéro du département. Nous récupérerons d'autre part le nom du département à partir de la table Départements. Enfin, nous compterons le nombre total d'incendies pour chaque département à l'aide de la fonction préétablie en SQL COUNT.

Après avoir sélectionné tous les termes que nous allons utiliser, il est temps de mettre en œuvre la jointure définie précédemment. Nous devons associer chaque incendie à son département en liant le numéro de département dérivé du code INSEE au code départemental dans la table Départements. De plus, nous allons regrouper les incendies par département afin d'obtenir un total pour chaque département.

Que gagne-t-on en faisant cette requête ?

En construisant cette table, nous avons fusionné deux tables indépendantes afin de centraliser les données souhaitées. Dans cette table , nous avions comptabilisé le nombre d'incendies par département sur le territoire français, nous permettant ainsi d'avoir une représentation plus claire du nombre d'incendies à l'échelle nationale. Par ailleurs, nous allons approfondir notre analyse dans la section Statistique de notre projet.

3.10 La Table Humidites

Afin d'étudier la question de l'impact de l'humidité sur les incendies, il est nécessaire de fusionner deux fichiers CSV.

Ainsi, nous avons deux méthodes : soit on les fusionne en utilisant le langage R, soit on utilise le langage Python et SQLITE3. Dans cette problématique on a decide d'utilise le language Python donc ce qu'on a effectue est qu'on a créer la Table Humidités avec les champs qu'on veut , et qui sont nécessaires pour les deux tables.

Après avoir établi la table, nous avons réalisé une autre opération pour introduire les informations des deux tableaux en ayant recours à une *INNER JOIN* sur les attributs désirés, en les fusionnant par l'égalité de code_INSEE entre ces deux ensembles de données.

Avec cette approche, nous obtiendrons une table qui regroupe les informations nécessaires, permettant ainsi une analyse plus aisée.

3.11 La Table Vents

Afin d'etudier la problematique qui se concentrent sur la Relation entre le vent et la propagation des incendies on a eu recours a la methode pour creer la Table Vents car on a besoin de fussionner deux fichiers CSV.

Ainsi, nous avons deux méthodes : soit on les fusionne en utilisant le langage R, soit on utilise le langage Python et SQLITE3. Dans cette problématique on a decide d'utilise le language Python donc ce qu'on a effectue est qu'on a créer la Table Vents avec les champs qu'on veut , et qui sont nécessaires pour les deux tables.

Après avoir établi la table, nous avons réalisé une autre opération pour introduire les informations des deux tableaux en ayant recours à une *INNER JOIN* sur les attributs désirés, en les fusionnant par l'égalité de code_INSEE entre ces deux ensembles de données.

Avec cette approche, nous obtiendrons une table qui regroupe les informations nécessaires, permettant ainsi une analyse plus aisée.

3.12 Initiation de la Carte de France

Dans le cadre de notre projet, nous traitons les incendies sur le sol français. Les données de cette unité d'enseignement contiennent la géolocalisation des incendies sur le territoire français.

Afin de représenter les incendies sur le territoire français de façon abstraite, nous avons utilisé le langage Python accompagné de plusieurs bibliothèques qui nous ont facilité l'élaboration de cette carte.

Dans cette partie de notre section informatique, nous détaillerons la manière dont nous avons réalisé cette carte.

Alors, pour initier, nous avons importé les bibliothèques :

1. **Geopanda** est une librairie pratique pour manipuler des données géospatiales telles que les fichiers GeoJSON et les cartes.
2. **Pandas** est une bibliothèque qui facilite la manipulation de données sous format tabulaire. Dans notre situation, elle nous donne la possibilité d'importer et de gérer les données relatives aux incendies.
3. **matplotlib.pyplot** est une librairie de représentation graphique de données qui facilite la création de graphiques, de cartes et divers autres éléments visuels. Dans notre situation, elle a été bénéfique pour élaborer la carte et y placer les repères des incidents.

Une fois que toutes les bibliothèques requises ont été importées, il faut procéder au chargement de la base de données, autrement dit, du fichier CSV. Le fichier que nous allons importer contient les coordonnées géographiques des incendies, soit la latitude et la longitude. Pandas lira ce fichier sous forme de **DataFrame**, aussi appelé **df**, dans une structure tabulaire.

Dans ce genre de contexte, nous aurons besoin de télécharger la carte de France, qui offre une vue d'ensemble incluant tous les départements. Nous avons obtenu cette carte depuis data.gouv.fr au format geojson. Afin d'importer ce fichier dans notre programme, nous spécifions le chemin d'accès contenant les contours des départements français.

Pour être plus précis, un fichier **GeoJson** est un format de données géographiques contenant des informations relatives aux formes géographiques et à leurs attributs, dans notre contexte, les départements français.

Suite au chargement de notre carte en format GeoJSON, nous attribuons une variable afin de permettre la lecture du fichier, ce qui nous amène à utiliser gdp.read_file. C'est l'une des techniques prédéfinies dans geopandas. Dans cette situation, les informations sont conservées dans un **geodataframe**, une structure de données conçue spécifiquement pour stocker des données spécialisées en matière d'informations géospatiales.

Après avoir rassemblé toutes les données nécessaires à l'élaboration de notre carte, nous avons intégré les fichiers contenant la localisation des incendies ainsi que le fichier **géospatial**.

Nous allons maintenant nous concentrer à la création du graphique et à la traçage de l'axe.

fig, ax = plt.subplots(dpi=150, figsize=(15, 15)) Cette ligne nous autorise à générer une figure et un axe pour le graphique de la carte grâce à **matplotlib**. L'argument figsize=(15,15) indique que la dimension de l'image sera de 15 pouces par 15 pouces.

Une fois la carte établie, nous devons débuter le traçage des départements en exploitant les données géospatiales. L'utilisation de la variable **gdf_departements**.plot nous donne la possibilité de représenter les départements sur la carte.

Nous tenons à préciser que **.plot()** est bénéfique pour visualiser les contours des départements à partir du **GeoDataFrame** sur l'axe que nous avons mis en place précédemment. En outre, l'option **color = « lightgray »** nous donne la possibilité de remplir en une teinte grise claire, tandis que l'option **edgecolor** nous autorise à indiquer que les bordures des départements sont dessinées en noir pour mieux marquer la délimitation de chaque département.

Après avoir délimité les départements sur la carte, il est nécessaire de marquer, ou autrement dit indiquer, les emplacements des incendies sur le sol français. Pour cela, nous utilisons en premier lieu **ax.scatter()** - l'une des méthodes de **Matplotlib** qui nous permet de créer un nuage de points (ou scatter plot en anglais) sur un graphique. Grâce à cette méthode, nous précisons via le DataFrame les deux colonnes que nous souhaitons utiliser comme longitude et latitude de notre fichier CSV. Ensuite, on précise que l'on a voulu

que la couleur des nuages ou des points soit représentée en rouge, symbolisant le feu ou les incendies, pour être plus précis.

On précise que l'option **s = 50** signifie que nous modifions la taille des points à 50 pixels, afin de montrer que cette valeur est élevée. Plus cette valeur augmente, plus les points seront grands sur la carte. Finalement, on précise que l'option **label** nous donne la possibilité de définir une légende en identifiant les points qui représentent des incendies.

Une fois les points ajoutés sur la carte, nous allons déterminer les contours de la carte française à cet endroit. Les limites territoriales de la France sont fixées par le Traité de Paris du 10 février 1947. De surcroît, depuis la fondation des Nations Unies en 1945, ces frontières sont reconnues au niveau international.

Ainsi, nous avons délimité les frontières de la France sur la base du traité de Paris du 10 février 1947. La longitude s'étend de -5.5 à 10 et la latitude de 41.5 à 51.5, en employant **ax.set_xlim** et **ax.set_ylim**.

Une fois les frontières du territoire français tracées, nous allons insérer un titre et une légende en employant **plt.title()** pour attribuer un titre à la carte avec une taille de police de 20 points. Par ailleurs, grâce à **plt.legend()**, nous allons également créer la légende qui associe le label 'Incendies' aux points rouges sur la carte pour signaler que ces derniers représentent des incendies.

Par la suite, nous allons retirer les axes en employant la technique **ax.set_axis_off()**.

Cette technique va éliminer les axes et les graduations sur les axes x et y afin de rendre la carte plus lisible et plus épurée.

Ensuite, nous avons deux dernières étapes : la sauvegarde de l'image et l'affichage de la carte. Pour la sauvegarde, nous avons recours à **plt.savefig()** en précisant le nom du fichier et son extension souhaités. Nous y avons également indiqué la résolution de l'image, spécifiant les dpi (points par pouce). Nous avons choisi 600, ce qui représente une qualité très élevée. De surcroît, nous avons activé l'option **bbox_inches = 'tight'**, ce qui élimine tous les espaces superflus autour de la carte pour la rendre plus compacte.

Pour afficher la carte, nous utilisons la fonction **plt.show()**. Cette technique nous autorise à présenter la carte, elle est employée pour en faire la visualisation.

4. Partie Statistique

4.1 Definitions des concepts statistiques

Avant de commencer à donner des définitions, il est essentiel de nous baser sur le concept initial, c'est-à-dire la définition du terme Statistique. On peut définir ou représenter la statistique comme l'ensemble des méthodes et techniques utilisées pour collecter, analyser, interpréter et présenter des données numériques.

Dans ce projet, nous allons nous concentrer sur la branche des statistiques connue sous le nom de Statistique Inférentielle.

La statistique inférentielle est une discipline des statistiques qui s'appuie sur les données d'un échantillon pour formuler des déductions ou effectuer des généralisations à propos d'une population plus vaste.

À l'opposé de la statistique descriptive qui se concentre sur le résumé ou la description des traits d'un ensemble de données, la statistique inférentielle offre la possibilité de réaliser des estimations et des tests concernant les paramètres d'une population basée sur des données issues d'un échantillon. Elle s'appuie sur la théorie des probabilités, ce qui rend possible des inférences rigoureuses et quantifiables.

Nous allons définir ci-dessus certains concepts statistiques que nous utiliserons dans notre analyse statistique.

Définition de la Moyenne

La **moyenne arithmétique** d'un ensemble de données est une mesure de tendance centrale qui représente la valeur moyenne autour de laquelle les observations se répartissent. Elle est définie comme le quotient de la somme des valeurs observées par le nombre total d'observations.

Soit un échantillon $X = \{x_1, x_2, ..., x_n\}$ de taille n, la moyenne $\bar{X}$ est donnée par :

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Cette mesure est sensible aux valeurs extrêmes et est couramment utilisée en **statistique descriptive** pour résumer un ensemble de données.

Définition de la Médiane

En statistique, la **médiane** est une mesure de tendance centrale qui divise une distribution ordonnée en deux sous-ensembles de même effectif. Elle est définie comme la valeur M telle que :

- **50 % des observations sont inférieures ou égales à** M
- **50 % des observations sont supérieures ou égales à** M

Mathématiquement, soit un échantillon de taille n constitué des observations $x_1, x_2, ..., x_n$ classées par ordre croissant :

- **Si** n **est impair** $(n = 2k + 1)$, la médiane est l'élément central :

$$M = x_{k+1}$$

- **Si** n **est pair** $(n = 2k)$, la médiane est la moyenne des deux valeurs centrales :

$$M = \frac{x_k + x_{k+1}}{2}$$

Définition de la Classe Modale

La **classe modale** est l'intervalle de valeurs qui renferme le plus fort effectif dans une distribution organisée en classes. Autrement dit, c'est la classe qui se manifeste le plus souvent dans un histogramme.

Définition de ggplot2

ggplot est une des librairies du langage R. Elle fait partie du package plotnine, qui facilite la création de graphiques.

Définition de la Population

En statistique, la population se réfère à l'ensemble de tous les individus, objets ou événements qui sont sujets à une étude.

Définition d'un Échantillon

Un échantillon représente une partie de la population étudiée. On fait appel à lui quand il est nécessaire d'analyser l'ensemble de la population en raison de sa complexité.

Définition d'une Variable

Une variable est un attribut quantifiable qui peut varier d'un individu à l'autre.

Définition d'une variable qualitative

Une variable qualitative représente une caractéristique ou une catégorie qui ne peut pas être quantifiée.

Définition d'une variable quantitative

Une Variable Quantitative représente une évaluation numérique et peut être l'objet d'opérations mathématiques.

4.2 Description des donnees

Avant de commencer l'étape d'analyse de nos données et leur présentation sous forme de graphiques, nous allons identifier les types de données fournies par notre équipe pédagogique, afin de mener à bien ce projet.

Donc pour cette UE on a eu 3 fichiers CSV contenant des donnees importants ou dans ce rapport on va se baser pour poser notre analyse.

Le fichier incendies renferme des attributs tels que le nom de la commune, le code de la commune, l'année de l'incendie, le mois de l'incendie, la date et l'heure à laquelle l'incendie a été signalé ainsi que les causes principales et secondaires de l'incendie. Ces informations nous seront utiles pour examiner nos données et caractériser ces phénomènes en fonction des problématiques que nous mettrons en place bientôt dans ce rapport.

Le fichier géographique renferme la latitude, la longitude et l'altitude. Ces informations peuvent nous donner l'emplacement précis du lieu de l'incendie, ce qui nous permettra de procéder à des analyses en posant nos questions.

Le dossier Météorologique nous fournira des caractéristiques liées aux conditions météorologiques, ce qui nous permettra de distinguer les divers types de météo au moment de l'incendie.

4.3 Analyse des doneees

Dans cette partie, une fois que toutes les définitions nécessaires sont établies et que notre base de données est complète, nous serons prêts à analyser les données grâce au langage R. Nous emploierons différents types d'histogrammes pour diversifier nos analyses et nous procéderons à une étude approfondie de chaque histogramme.

Nous avons divisé nos problématiques en différentes sections :

4.3.1 Évolution des incendies

4.3.1.1 Évolution des incendies au fil des années Dans cette problématique, on va interroger sur le nombre d'incendies qui se produisent chaque année sur le territoire français. Pour répondre à cette question, il est primordial de définir d'abord certains concepts naturels qui faciliteront l'analyse de cette problématique. Une année se compose de 12 mois et comporte 365 jours. Dans cette étude, nous examinerons l'évolution annuelle du nombre d'incendies à travers tous les départements français, déterminant s'il est en déclin ou en ascension.

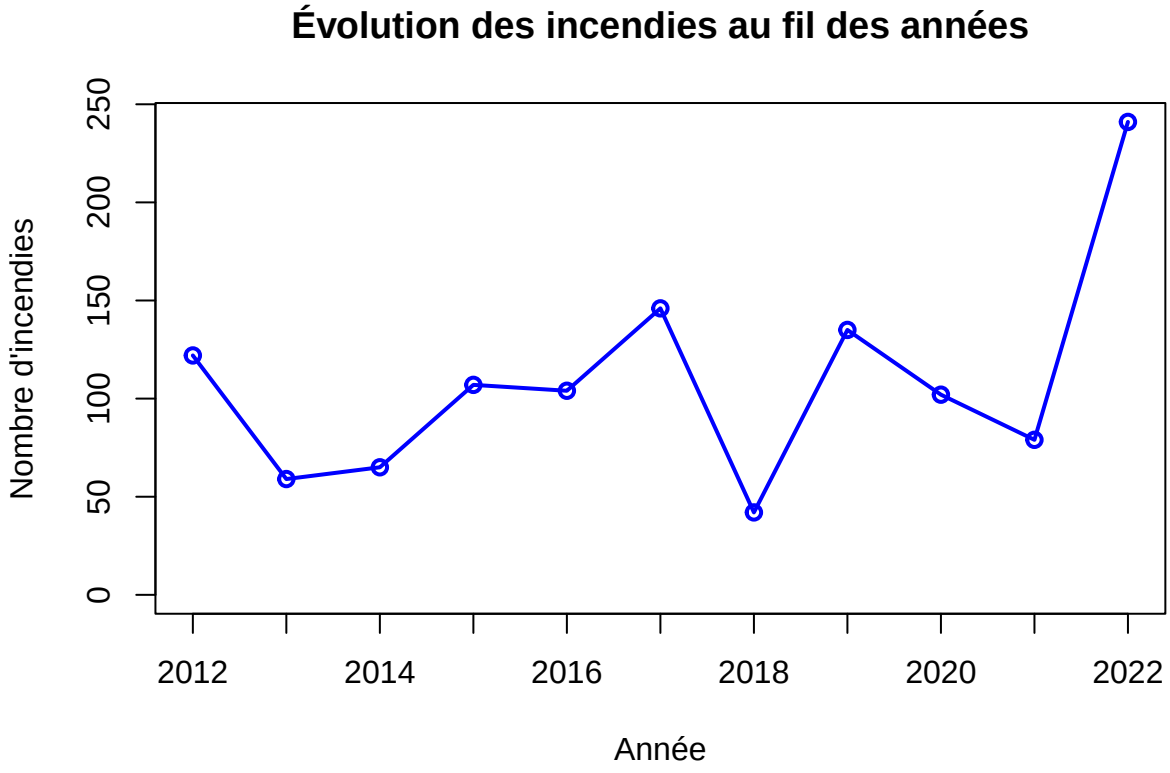
Pour réaliser cette analyse statistique, il faut utiliser la table des **incendies**.Pour réaliser l'histogramme, nous devons utiliser le langage R. Nous avons précisé le chemin d'accès au fichier CSV grâce à une variable que nous avons définie, une variable donnée qui va recevoir la fonction read.csv et le chemin du fichier. Cela permettra à la variable d'accéder au fichier CSV. De plus, pour vérifier notre travail, nous utiliserons la méthode head qui nous donnera les six premières entrées de notre fichier CSV afin de nous assurer que nous travaillons sur le bon fichier et de vérifier également l'en-tête avec les attributs à utiliser.

Pour réaliser cette analyse, nous avons adopté cette technique qui consiste à déterminer la fréquence des incendies par année. En d'autres termes, nous comptabilisons le nombre d'incendies pour chaque année dans la colonne « année » du jeu de données. Cela va nous permettre de créer un tableau qui associe chaque année à son nombre d'incendies.

De plus, nous avons élaboré ce genre de graphique en utilisant un graphique linéaire. Nous avons exploité la fonction « plot() » avec les fréquences obtenues. L'option « o » a été utilisée pour indiquer que nous allions visualiser l'évolution des incendies à la fois par des points et des lignes sur la période donnée.

Pour adapter notre graphique à nos besoins, nous avons opté pour la couleur bleue afin de le rendre plus personnalisable et lisible. De plus, nous avons défini les titres des axes du graphique en utilisant les paramètres xlab, ylab et main.

```
donnees <- read.csv("../Data/donnees_incendies.csv")

annee_freq <- table(donnees$annee)

plot(annee_freq,
     type="o",  # "o" pour un graphique avec des points et des lignes
     col="blue",
     main="Évolution des incendies au fil des années",
     xlab="Année",
     ylab="Nombre d'incendies")
```



1. **Analyse Informatique:**

Nous allons détailler le processus de développement du code pour structurer notre graphique. Pour cela, nous avons importé les données du fichier grâce à la méthode **read.csv()**. Dans cette méthode, nous avons spécifié le chemin relatif du fichier csv.Par la suite, nous avons déterminé la fréquence des incendies par an en optant pour la colonne **donneees$annee**.En employant la méthode **table()**, nous dénombrons le nombre d'incendies survenant par année.

Cela nous autorisera à constituer un tableau où les indices représentent les années et les valeurs correspondent aux occurrences des incendies.

À présent, nous passons à la création du graphique à l'aide de la méthode **plot()**. Nous avons indiqué la variable **annee_freq** et dans cette méthode, nous avons précisé le type 'o' pour indiquer que les points doivent être affichés et reliés par une ligne. De plus, nous avons défini l'option couleur afin de spécifier que le graphique doit être en bleu.

Avec l'option **main**, nous désignons le titre du diagramme. Ensuite, grâce à xlab et ylab, nous définissons les étiquettes des axes x et y, où x représente l'**année** et y représente le **nombre d'incendies**.

2. **Analyse Statistique:**

Alors, débutons par cette étude ; nous allons mener notre analyse sur une période de 11 ans, de 2012 à 2022.

Nous allons donc commencer à dresser le bilan de nos effectifs d'incendies. En 2012, nous avons enregistré 122 incidents, en 2013, 59 incendies, en 2014, 65 incendies, en 2015, le nombre a grimpé à 107. En 2016, nous avons eu 104 incendies. Pour l'année suivante, en 2017, nous avons connu une augmentation avec 146 incidents. Puis en 2018, le chiffre est redescendu à 42 incendies. En 2019, nous avons rapporté 135 incendies et pour l'année suivante, en 2020 nous avions enregistré 102 incidents. Enfin pour2021 ,nous avons comptabilisé 79 incendies et pour2022 ,le nombre d'incendies a fortement augmenté atteignant 241incendies .

Il est possible d'observer que la moyenne générale des incendies en France sur une durée de 11 ans s'élève à 109,27. Maintenant que nous avons compilé les totaux des onze années consécutives, nous allons détailler notre analyse. En démarrant de 2012, nous constatons un total de 122 incendies, ce qui pourrait être considéré comme « bon ». Cependant, cette comparaison est prématurée tant que nous n'avons pas répertorié les autres chiffres pour une évaluation plus approfondie. Après une baisse de 63 incendies en 2013, on a constaté une augmentation de 6 incendies en 2014 lors de la comparaison entre 2013 et 2014.

Cependant, en 2015, le nombre d'incendies a considérablement augmenté, atteignant 107 incendies pour l'année 2015, ce qui représente un chiffre élevé mais pas le plus élevé que nous ayons connu. En comparaison avec 2012, nous avons atteint un taux qui est le deuxième record maximal. Puis en 2016, ce taux a baissé de trois incendies par rapport à l'année précédente.

En 2017, on a observé une hausse remarquable du taux d'incendies qui a atteint un nouveau sommet à 146, soit une augmentation de 42 incendies par rapport à 2016. Si l'on compare avec l'année 2012, cela représente une différence de 24 incendries. En 2018, on constate une baisse rapide du nombre d'incendies, enregistrant un taux de 42 qui, selon les données de 2018, représente le plus bas enregistré depuis 2012. En 2019, on a noté une hausse significative du nombre d'incendies, atteignant 135, soit une différence de 93 incendies par rapport à 2018.

En 2020, une baisse du nombre d'incendies entraîne une variation de 33 incendies par rapport à l'année 2029. Ensuite, on observe une baisse en 2021 avec une différence de 23 incendies par rapport à l'année 2020. Finalement, un chiffre très élevé en 2022 a été atteint, atteignant le nombre record de 241 incendies. De plus, cela représente une différence de 162 incendres par rapport à l'année 2021.

On peut résumer qu'au cours de ces 11 années, l'année où le nombre d'incendies est le plus élevé est 2022 avec un total de 241 incendies, tandis que l'année où ce nombre est le plus bas est 2018.

En 2022, on a observé une augmentation atypique, avec un total de 241 incendies. Divers éléments peuvent expliquer cette augmentation, tels que les modifications extrêmes des conditions météorologiques. En 2022, des vagues de chaleur sévères et une sécheresse persistante ont contribué à cet accroissement. De plus, les actions humaines, notamment la négligence, ont influencé cette hausse.

Pour conclure notre analyse, l'évolution des incendies de 2012 à 2022 a montré des tendances significatives et atypiques. Cela souligne l'importance d'examiner l'impact des facteurs climatiques, humains et socio-économiques sur le nombre d'incendies. On peut conclure que l'année 2022 a été exceptionnelle, affichant le plus haut nombre d'incendies. De plus, nous avons observé une variabilité des tendances, comme démontré souvent avec les années 2017, 2014 et 2013 où les taux étaient remarquablement bas comparés à ceux de 2022. Il est essentiel de noter que le changement climatique a une grande importance.

Enfin, et c'est crucial, la gestion et la prévention jouent un rôle primordial. Les préfectures, ainsi que les forces de police municipales et nationales, peuvent contribuer à maîtriser ce taux d'incendies attribués à des actes de malveillance ou de négligence.

4.3.1.2 Analyse des heures d'incendie

4.3.1.3 Évolution décennale des causes d'incendies

4.3.1.4 Heures critiques de déclenchement

4.3.1.5 Cyclicité hebdomadaire

4.3.2 Facteurs climatiques et météorologiques

4.3.2.1 Influence de la température sur les incendies

4.3.2.2 Impact de l'humidité sur les incendies Pour mener l'analyse de l'influence de l'humidité sur les incendies, il est nécessaire de constituer notre base d'analyse. Il est nécessaire d'utiliser les deux tables que nous avons mises en place dans notre Base de données, à savoir la Table Incendies et la Table donnees_meteo.

Afin de réaliser une jointure entre ces deux tables, nous avons fait appel à une troisième table nommée humidite qui regroupe les champs des deux tables visées, partageant un élément en commun : le « Code_INSEE ». Nous avons détaillé la méthode utilisée pour cela dans la section Informatique de notre rapport.

Avant d'approfondir dans les détails de notre étude, nous allons d'abord définir les termes clés que nous utiliserons dans notre analyse. L'humidité se réfère à la présence de vapeur d'eau dans l'air.

Le but de notre analyse est d'étudier le lien entre l'humidité atmosphérique et les incendies. Pour accomplir cela, nous devons étudier la relation entre l'humidité et la dimension des incendies.

Il est essentiel de mettre en évidence deux attributs importants.

1. **Tens_vap_med**: Cet attribut évalue la pression de vaporisation moyenne, qui est une autre façon de dire qu'il s'agit d'un indicateur de l'humidité de l'air.
2. **surface_parcourue_m2**: Cette caractéristique présente la superficie ravagée par un feu, ce qui en fait un instrument pour évaluer l'intensité du feu.

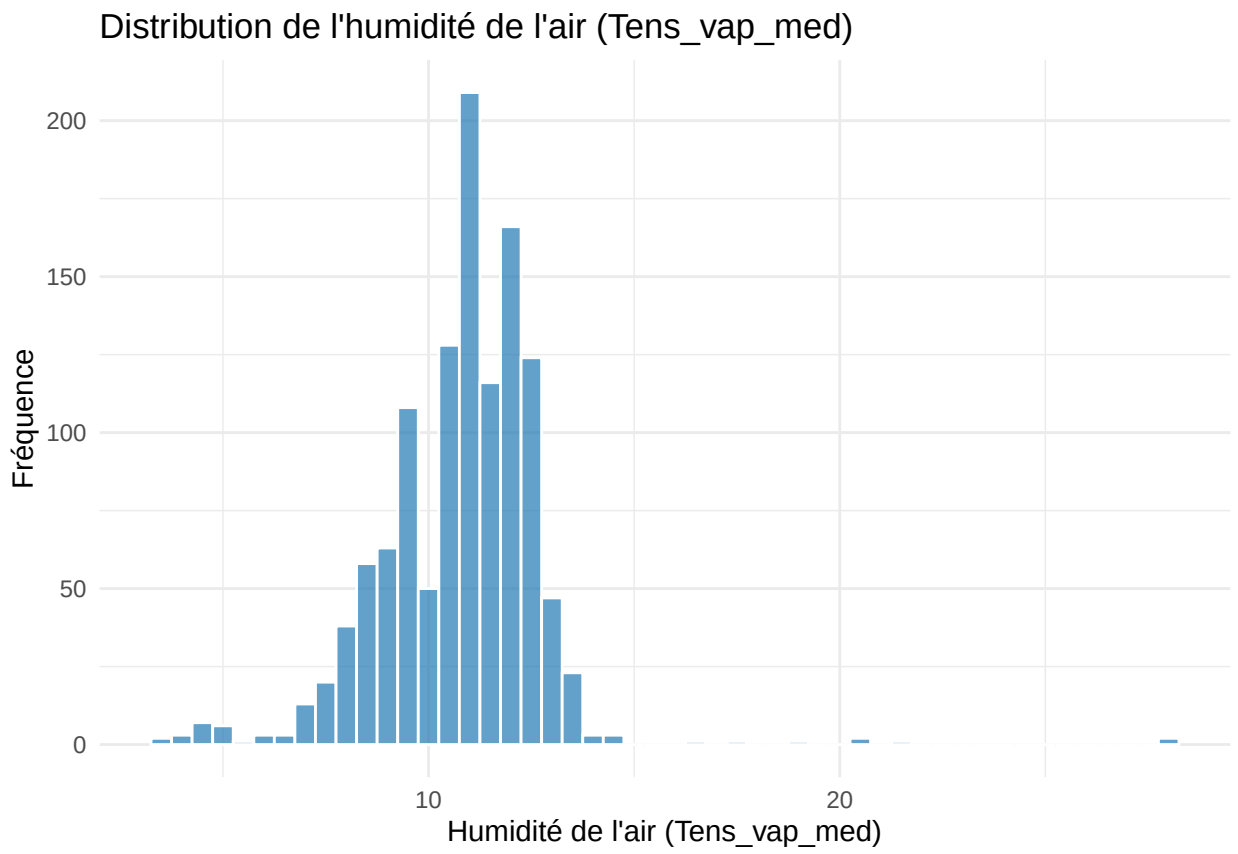
Dans cette étude, nous allons créer et analyser deux graphes indispensables à notre problématique.

1. **Histogramme de l'humidité**

Le diagramme de l'humidité nous aidera à examiner et à saisir la distribution des taux d'humidité dans l'échantillon de données.

L'humidité joue un rôle crucial dans l'analyse des incendies en raison de son impact sur la rapidité de leur avancement. Cependant, avant d'examiner toute relation avec la surface incendiée, nous devons d'abord comprendre comment l'humidité fluctue dans les données.

```r
library(ggplot2)
data <- read.csv("../Exports/export_Humidites.csv")
ggplot(data, aes(x = Tens_vap_med)) +
  geom_histogram(binwidth = 0.5, fill = "#1f77b4", color = "white", alpha = 0.7) +
  labs(title = "Distribution de l'humidité de l'air (Tens_vap_med)",
       x = "Humidité de l'air (Tens_vap_med)",
       y = "Fréquence") +
  theme_minimal()
```



1. **Analyse Informatique:**

Dans notre code destiné à la création du graphique, nous avons employé le langage R pour sa réalisation. Tout d'abord, nous avons importé le fichier en utilisant la méthode **read.csv()**.

Par la suite, on procède à l'initialisation du graphique que l'on souhaite concevoir avec la méthode **ggplot()**.On passe en paramètres de cette méthode les données ainsi que l'**aes**,qui nous permet de spécifier que l'axe des x sera représenté par **Tens_vap_med**.

On définit ensuite une autre méthode **geom_histogram()** pour intégrer un histogramme au graphique. Dans cette méthode, on spécifie des paramètres pour déterminer la largeur des barres de l'histogramme, et par la suite, on remplit les barres en bleu. Nous définissons la bordure en blanc et rendons les barres davantage transparentes.

On détermine les titres et étiquettes en faisant appel à la méthode **labs()**, en exploitant les paramètres title, x et y.Le titre indique le sujet du graphique, 'x' correspond à l'étiquette de l'axe horizontal et 'y' correspond à l'étiquette de l'axe vertical.

Et pour donner un aspect minimaliste au thème, nous avons employé la méthode **theme_minimal()** afin d'incorporer un style plus contemporain.

2. **Analyse Statistique:**

L'axe des abscisses illustre l'humidité de l'air, déterminée par la pression de vapeur moyenne. Les chiffres se situent approximativement entre 0 et 25.On présume que les valeurs sont exprimées en hPa. C'est une unité utilisée pour évaluer la pression de la vapeur.

L'axe ordonnes illustre la fréquence, c'est-à-dire le total des observations pour chaque plage de tension de vapeur. Il a atteint une fréquence maximale de 200.

D'après les informations intégrées, cet histogramme révèle une distribution asymétrique avec une importante concentration de valeurs faibles en matière d'humidité de l'air, allant de 5 à 15 hPa.

L'histogramme présente une asymétrie vers la droite, avec un grand nombre d'observations ayant de faibles valeurs de tension de vapeur, indiquant une humidité faible à modérée, et quelques observations à des valeurs plus élevées, traduisant une humidité supérieure.

On pourrait affirmer que la classe modale de cet histogramme se situe approximativement entre 10 et 12 hPa.

La portée de cet histogramme s'étend de 0 à 25 hPa. Néanmoins, on remarque qu'il existe très peu de données entre 20 et 25 hPa. De plus, il est évident qu'au-delà de 22 hPa, aucune observation n'est présente.

Dans notre histogramme, on peut observer la présence d'une longue queue, bien qu'elle soit peu dense. Autrement dit, cela nous indique que les valeurs élevées de tension de vapeur sont peu fréquentes dans cet ensemble de données.

Nous allons déterminer les interprétations des intervalles concernant le taux d'humidité.

1. **0 a 5 hPa**: Cette plage indique un taux d'humidité très bas.
2. **5 a 15 hPa**: C'est la zone où est rassemblée la plupart des données. La fréquence s'accroît rapidement dès que l'on atteint 5 hPa, atteignant un maximum aux alentours de 10 à 12 hPa avant de redescendre. Cela signifie que le niveau d'humidité est modéré.
3. **15 a 20 hPa**:Dans cette période, nous observons une réduction qui est associée à des conditions plus humides.
4. **20 a 25 hPa**:Les observations sont peu fréquentes. La fréquence étant pratiquement de 0, on peut observer que nous avons des conditions hors du commun, telles que des climats tropicaux.

Dans un cadre météorologique, la tension de vapeur représente une évaluation de la pression partielle de la vapeur d'eau dans l'atmosphère, liée directement à l'humidité. Selon notre histogramme, une pression de vapeur de 10 hPa est associée à une humidité relative modérée, tandis qu'une pression de vapeur avoisinant les 20 hPa indique un niveau d'humidité considérablement plus élevé, potentiellement proche de la saturation.

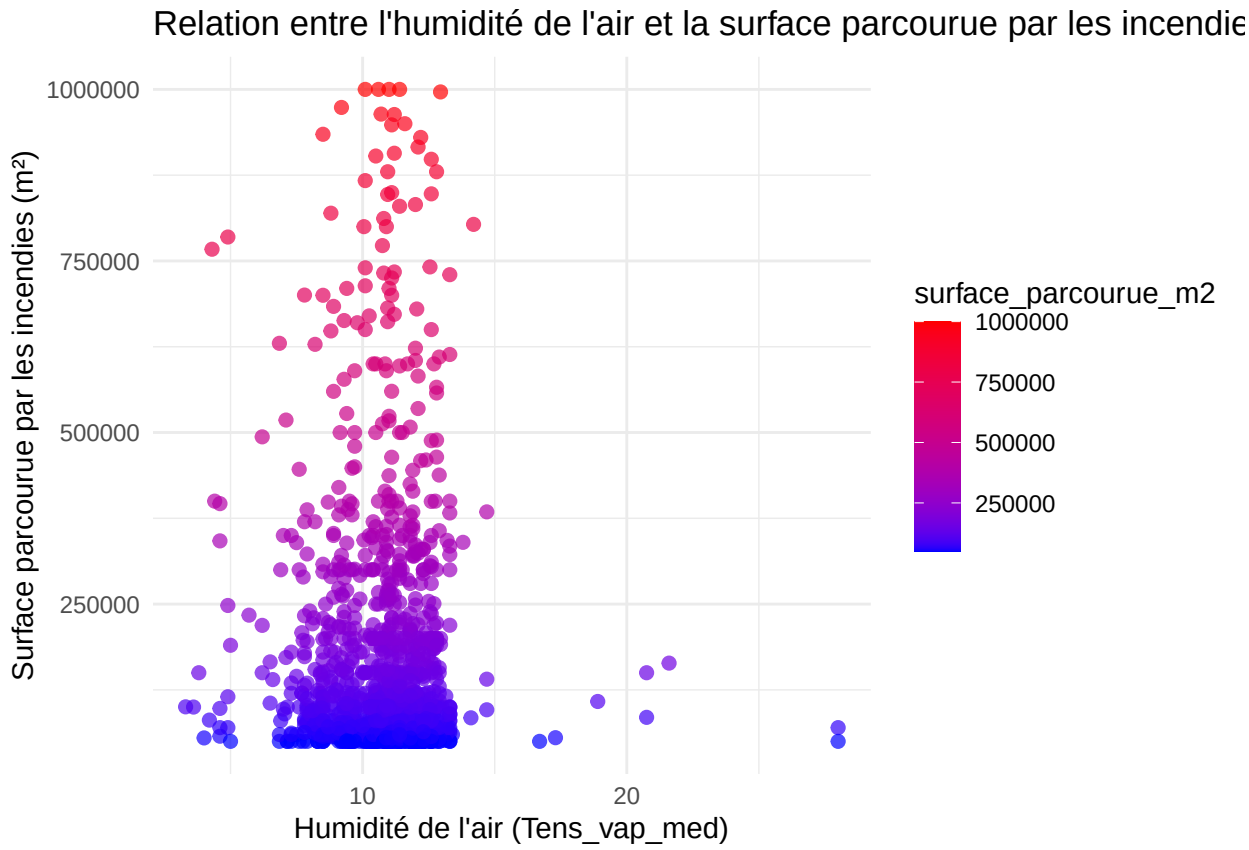
Cet histogramme indiquant une concentration autour de 10-12 hPA suggère un climat tempéré, caractérisé par une humidité généralement modérée la majorité du temps, mais avec des périodes plus humides de manière sporadique.

2. **Diagrame de dispersion**

Ce schéma nous offre la possibilité d'examiner s'il y a une relation entre l'humidité atmosphérique et l'étendue des incendies (autrement dit, la superficie qu'ils couvrent).

L'objectif est d'observer comment l'humidité influe sur la taille des incendies de manière perceptible.

```r
library(ggplot2)
data <- read.csv("../Exports/export_Humidites.csv")
ggplot(data, aes(x = Tens_vap_med, y = surface_parcourue_m2)) +
  geom_point(aes(color = surface_parcourue_m2), size = 2, alpha = 0.7) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "Relation entre l'humidité de l'air et la surface parcourue par les incendies",
       x = "Humidité de l'air (Tens_vap_med)",
       y = "Surface parcourue par les incendies (m²)") +
  theme_minimal()
```



Dans ce diagramme, seul l'axe des X représente l'air mesuré par la tension de vapeur moyenne. Les valeurs varient de 0 à 25 hPa. L'axe des Y représente la superficie parcourue par les incendies, exprimée en mètres carrés.

Nous avons choisi d'utiliser un graphique de type nuage de points, également appelé **scatter plot**, où chaque point représente une observation (incendie) associée à deux variables. Premièrement, il s'agit de l'humidité atmosphérique au moment du sinistre, et en second lieu, de la superficie affectée par cet incendie (Y).

La couleur des points varie du bleu au rouge, conformément à l'échelle indiquée sur la droite. Les points bleus représentent des surfaces inférieures à 250 000 m², alors que les points rouges sont associés à des surfaces supérieures.

Débutons par l'étude de la répartition des points, en mettant d'abord l'accent sur leur concentration. La plupart des points se regroupent dans la plage d'humidité de 0 à 15 hPa, avec une densité particulièrement élevée entre 5 et 12 hPa. Cela entraîne des niveaux d'humidité modérés.

On remarque également que le nombre de points au-delà de 20 hPa est très limité, ce qui suggère que les incendies dans des conditions extrêmement humides sont rares.

En ce qui concerne la superficie affectée par les incendies, allant de 0 à 250 000 m², on remarque que la majorité d'entre eux ont une portée assez restreinte. Pour les surfaces allant jusqu'à 1 000 000 m2, le graphique semble indiquer que les incendies de plus grande ampleur sont moins communs.

Concernant le lien entre l'humidité et la surface parcourue, on remarque une concentration de points bleus dans l'intervalle d'humidité de 5 à 15 hPa. Cela indique que les incendies de faible envergure se produisent plus souvent dans des conditions d'humidité modérées. En ce qui concerne les points rouges, ils sont plus éparpillés et se situent dans la même fourchette, bien qu'il existe des points rouges dans des zones où l'humidité est à la fois plus basse et plus élevée.

On ne constate pas de lien clair et direct entre le taux d'humidité de l'air et la superficie touchée par les feux. Les feux de grande envergure (indiqués par des points rouges) surviennent à divers niveaux d'humidité, toutefois, la plupart d'entre eux (qu'ils soient petits ou grands) se regroupent dans l'intervalle de 5 à 15 hPa.

Cependant, une tendance mineure peut être observée : les incendies de plus grande envergure (près de 1 000 000 m²) ont l'air de survenir un peu plus fréquemment dans des conditions d'humidité plus basse (environ 5 hPa ou moins), où l'air est plus sec, ce qui facilite la diffusion des flammes. Néanmoins, cette tendance n'est pas très prononcée.

3. **Comparaison avec l'histograme precedent:**

L'histogramme précédent nous indiquait que la pression de vapeur moyenne se situait approximativement entre 5 et 15 hPa, avec un sommet autour de 10-12 hPa. Cette distribution est confirmée par ce nuage de points.

Les quelques rares points au-delà de 20 hPa dans l'histogramme témoignent de la confirmation que les incendies sont peu fréquents dans des conditions très humides.

4. **Analyse Statistique**

Après avoir réalisé l'analyse des deux graphiques, nous sommes en mesure d'effectuer une analyse statistique.

Nous allons effectuer un calcul de corrélation.Elle nous offrira la possibilité d'évaluer l'intensité et le sens de la corrélation linéaire entre l'humidité atmosphérique et les dimensions des feux.

Avant tout, définissons les choses. Il s'agit d'une mesure statistique qui illustre la force et la direction d'un lien entre deux variables. Elle nous aide, de manière simple, à saisir comment deux variables se déplacent l'une par rapport à l'autre. On utilise le coefficient de corrélation de Pearson, qui se situe entre -1 et 1, pour quantifier la corrélation. Avec 1 représentant une corrélation parfaitement positive, -1 une corrélation parfaitement négative et 0 signifiant aucune corrélation.

```
correlation <- cor(data$Tens_vap_med, data$surface_parcourue_m2, use = "complete.obs")
print(paste("Corrélation entre Tens_vap_med et surface_parcourue_m2: ", correlation))
```

```
## [1] "Corrélation entre Tens_vap_med et surface_parcourue_m2:  -0.0211442157372533"
```

Dans ce code, nous avons fait appel à la fonction « cor() » afin de déterminer la corrélation de Pearson entre Tens_vap_med et surface_parcourue_m2, en omettant les variables manquantes. La méthode cor() nous donnera un coefficient de corrélation, comme expliqué précédemment.

1. Si la corrélation est haute, proche de 1, cela nous permet d'affirmer qu'une hausse de l'humidité est liée à une extension des incendies.
2. Si la corrélation est négative (proche de -1), cela implique qu'une hausse de l'humidité est liée à une réduction de la grandeur des feux.

3. Une corrélation proche de 0 suggère l'absence d'une relation linéaire manifeste.

L'analyse de corrélation révèle qu'il n'existe pas de lien linéaire prononcé entre le taux d'humidité et l'ampleur des feux dans vos informations. Selon cette étude, l'humidité semble avoir une influence marginale sur l'ampleur des incendies.

4.3.2.3 Relation entre le vent et la propagation des incendies Pour examiner la question relative à la corrélation entre le vent et la diffusion des incendies sur le sol français, nous adopterons une approche distincte en segmentant ce sujet en quatres sous-questions.

C'est pourquoi, pour aborder ce problème, on devrait considérer ces cinq enjeux :

1. Force du vent
2. Surface parcourue par le feu en fonction de la Force du vent
3. Force du vent par zone géographique
4. Surface parcourue par le feu par zone géographique

Avant d'examiner les quatre sous-problèmes, nous allons inspecter la corrélation.Nous pourrions vérifier la corrélation entre la puissance du vent et la surface parcourue en mètres carrés. En procédant ainsi, nous pourrions déterminer s'il existe une relation linéaire entre ces deux variables.

On remarque donc que si le coefficient de corrélation se rapproche de -1 ou 1, cela signifie qu'il existe une relation linéaire significative entre ces deux variables.

```
data <- read.csv("../Exports/export_vents.csv")
cor(data$Force_vent_med, data$surface_parcourue_m2, use = "complete.obs")
```

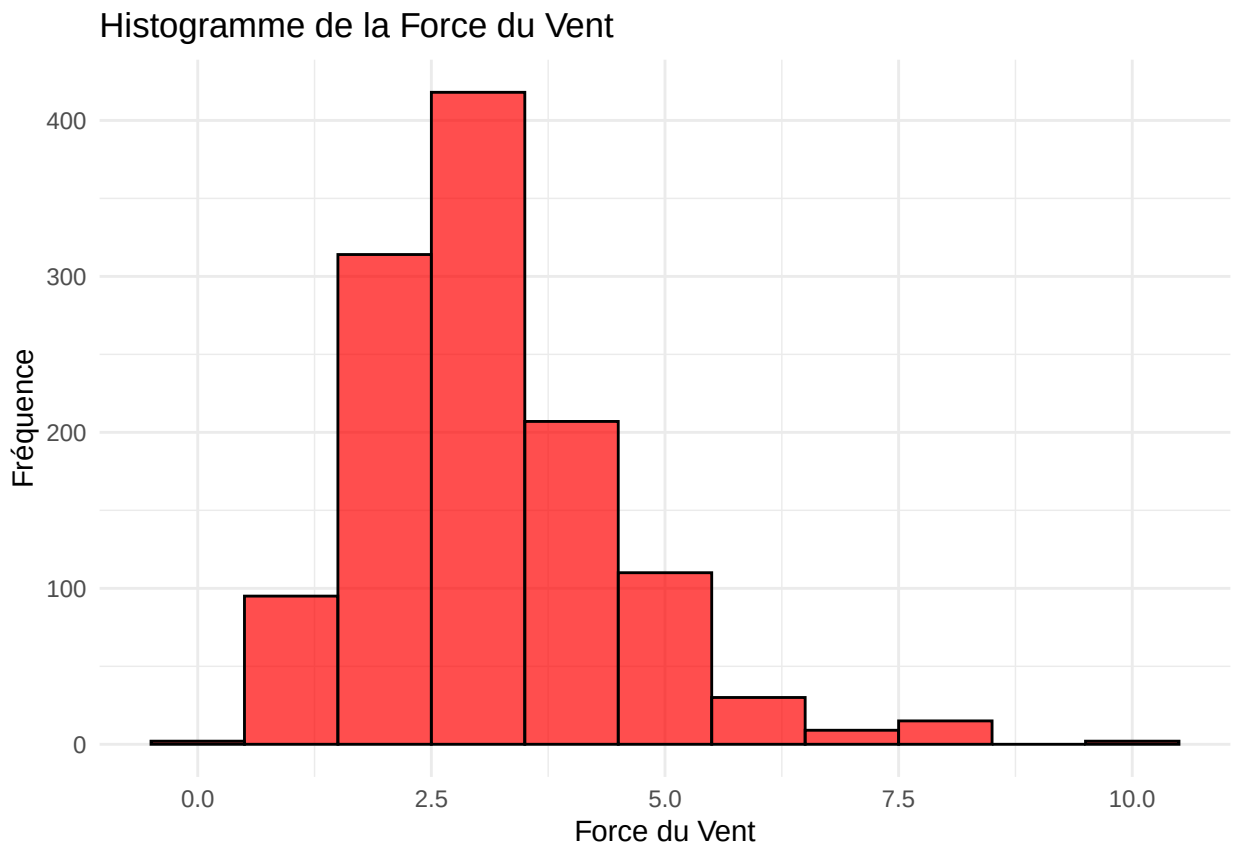
```
## [1] 0.03137438
```

Puisque le coefficient de corrélation de Pearson est proche de 0, cela indique qu'il n'existe pas de relation linéaire significative entre les deux variables.

1. **Force du vent**

Dans cette sous-problematique on va analyser la force du vent:

```
library(ggplot2)
data <- read.csv("../Exports/export_vents.csv")
ggplot(data, aes(x = Force_vent_med)) +
  geom_histogram(binwidth = 1, fill = "red", color = "black", alpha = 0.7) +
  labs(title = "Histogramme de la Force du Vent", x = "Force du Vent", y = "Fréquence") +
  theme_minimal()
```



1. **Analyse Informatique:**

Tout d'abord, nous allons détailler la méthode de construction de notre graphique. Nous avons utilisé la bibliothèque ggplot du langage R pour élaborer cet histogramme.Ensuite, nous avons importé nos données en spécifiant le chemin relatif du fichier CSV que nous avons conçu et développé à l'aide du langage Python et de la gestion des fichiers.

Dans le processus de chargement du fichier, nous avons employé la méthode read.csv pour interpréter le fichier CSV, et nous avons enregistré ces informations dans la variable data.

Par la suite, nous avons fait appel à la méthode, définie dans la bibliothèque ggplot2, soit la méthode ggplot. Nous lui avons précisé l'emplacement des données stockées et mis en place une autre méthode aes, signifiant **aesthetics**, pour indiquer quelles colonnes de données devraient être visualisées sur les axes. Dans notre situation, nous attribuons l'axe des x à la variable Force_vent_med qui va symboliser l'intensité du vent mesurée.

Par la suite, nous employons la méthode **geom_histogram()** pour intégrer l'histogramme à la représentation graphique. Nous spécifions la largeur des barres, la teinte que nous voulons utiliser pour leur remplissage, la nuance du contour et également le degré de transparence des barres.

On termine par la définition des étiquettes et des titres de notre histogramme grâce à la méthode **labs()**. On y inclut le titre du graphique ainsi que les libellés pour les deux axes, x et y. De plus, comme nous avons opté pour un style minimaliste, nous avons utilisé la méthode **theme_minimal()** afin de conférer un aspect plus contemporain au graphique.

2. **Analyse Statistique:**

On présume que la force du vent est évaluée en kilomètres par heure.

Cette histograme represente la distribution de la force du vent mesure en Km/h. L'échelle horizontale montre la puissance du vent, avec des valeurs variant de 0 à 10,5.L'axe des y symbolise la fréquence, soit le nombre d'apparitions de chaque plage de force du vent, avec des valeurs se situant approximativement entre 0 et 400.

L'histogramme indique une distribution asymétrique vers la droite, ce qui implique une concentration accrue de données pour des valeurs faibles de la force du vent, avec une longue traîne vers les valeurs plus élevées.

Le sommet de l'histogramme se trouve approximativement dans la gamme de 3.5 à 4.0 pour la force du vent, avec une fréquence avoisinant les 400. Cela nous indique que dans cet échantillon, la force du vent la plus courante se situe dans cette plage.

Concernant la portée des valeurs, la force du vent varie approximativement de 0 à 10,5. Toutefois, les occurrences de valeurs dépassant 8 sont extrêmement rares, ce qui indique que de très forts vents sont peu fréquents dans cet ensemble de données.

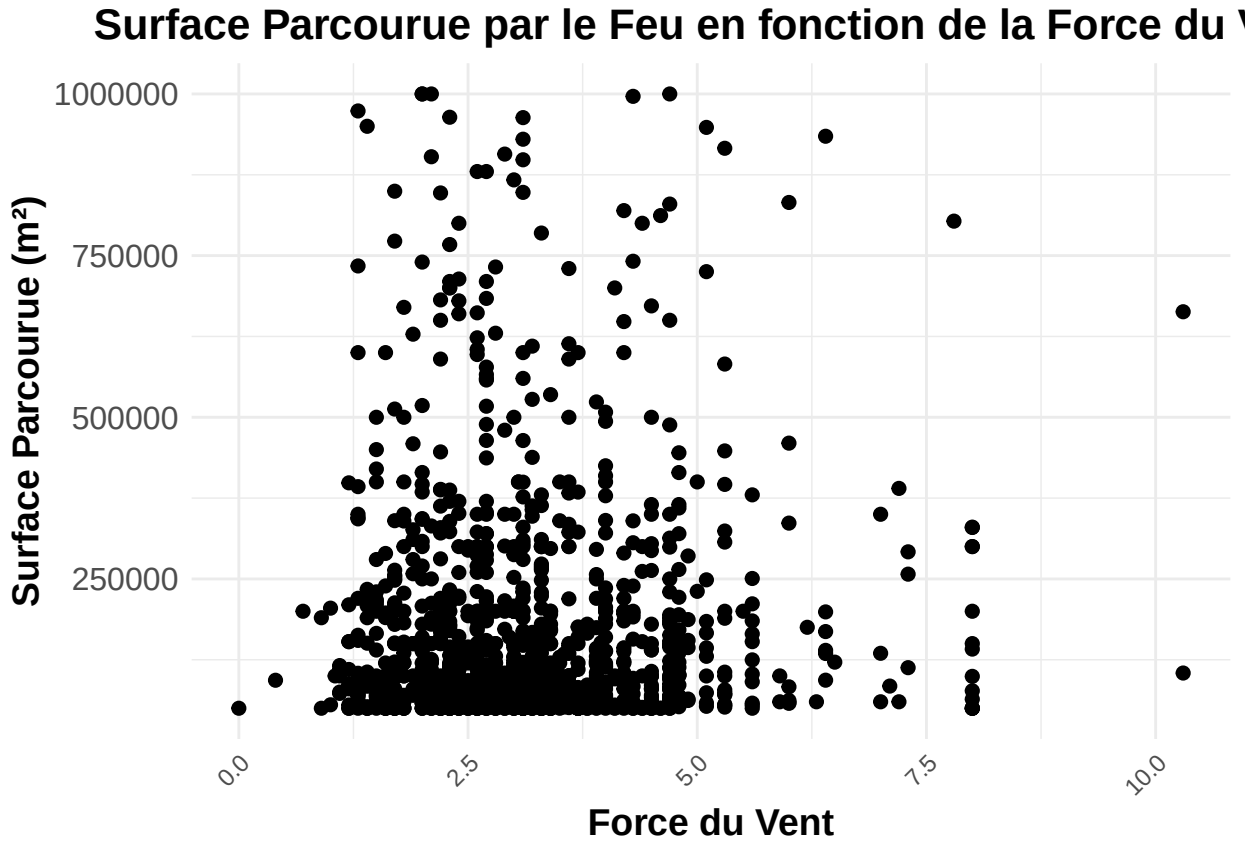
Presque 80% des données se regroupent entre 0 et 5.5 environ, ce qui indique que cet échantillon est principalement dominé par les vents légers à modérés.

La moyenne de cet histogramme se situe légèrement au-dessus du mode, estimée approximativement entre 4.0 et 4.5.Concernant la médiane, elle se positionne probablement autour de 3.5, car la distribution penche vers la droite, divisant l'échantillon en deux segments égaux.

Cet histogramme pourrait illustrer des relevés de la puissance du vent sur une période déterminée.La prévalence de brises légères à modérées (0.0 à 5.5) indique un climat plutôt paisible, avec la présence exceptionnelle de vents puissants au-delà de 8, qui pourraient survenir lors d'événements rares associés à des phénomènes météorologiques tels que des tempêtes.

2. **Surface parcourue par le feu en fonction de la force du vent**

```r
library(ggplot2)

data <- read.csv("../Exports/export_vents.csv")

ggplot(data, aes(x = Force_vent_med, y = surface_parcourue_m2)) +
  geom_point(color = "black", size = 2) +
  labs(title = "Surface Parcourue par le Feu en fonction de la Force du Vent",
       x = "Force du Vent",
       y = "Surface Parcourue (m²)") +
  theme_minimal() +  # Thème minimal pour une présentation épurée
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.text.y = element_text(size = 12),
    axis.title = element_text(size = 14, face = "bold"),
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5)
  )
```



1. **Analyse Informatique:**

Tout d'abord, nous allons décrire comment nous avons développé notre code. Pour commencer, nous avons importé notre bibliothèque ggplot2 afin de créer des visualisations complexes en langage R. Par la suite, nous avons chargé notre fichier contenant les données grâce à la méthode **read.csv()** et avons assigné le résultat à une variable nommée data. Cette variable data représente l'objet où nous conservons ces informations.

Ensuite, nous élaborons notre diagramme de dispersion en utilisant la méthode **ggplot()**, qui intègre les données. Par ailleurs, l'axe des x, déterminé par l'aes, représentera la force du vent alors que l'axe des y illustrera la surface parcourue en m^2.

Actuellement, nous sommes dans la phase où nous ajoutons des points pour pouvoir représenter les données. Nous utilisons la méthode **geom_point()** pour ce faire. Nous avons spécifié la couleur des points en noir et aussi déterminé la taille des points pour une visibilité optimale en utilisant l'option size = 2.

Pour une lecture optimale du graphique, nous avons utilisé la fonction **labs()**. Nous avons ajouté le titre au graphique et précisé les noms des deux axes en recourant à x et y.

Comme le graphe precedent on a utilise la methode du **theme_minimal()** pour le mettre dans un desin plus minimaliste

À l'instar du graphique précédent, nous avons employé la technique du **theme_minimal()** afin de le présenter dans une design plus épuré.

Pour accroître la clarté des axes et du titre, nous faisons appel à la fonction **theme()**. En manipulant **axis.text.x** et **axis.text.y**, nous avons employé l'option **element_text** pour le rendre incliné, modifier sa taille, etc. Quant aux attributs des deux axes, nous avons utilisé axis.title pour mettre le texte en gras grâce à l'attribut **face**.Pour le titre principal du graphique, on utilise **plot.title** pour ajuster ses caractéristiques à l'aide de **size**, **face** et **hjust**.L'utilisation de **hjust** permet de positionner le titre au centre.

2. **Analyse Statistique:**

Commençons par examiner ce sous-problème qui se focalise sur la superficie que couvre le feu en relation avec l'intensité du vent. Tout d'abord, on remarque que la force du vent varie entre 0 et 10 unités. De plus, la superficie affectée par le feu ou les incendies est presque de 0 à 1 000 000.

On remarque une concentration significative de points entre 0 et 5 unités de force du vent, avec une superficie allant de 0 à environ 750 000 m^2.

Il est également possible de conclure qu'au-delà de cinq unités, les points commencent à devenir rares et la surface parcourue tend à se stabiliser tout en réduisant légèrement.

L'accumulation de points indique qu'une intensification du vent a tendance à élargir la portée du feu, particulièrement dans le cas de vents faibles à modérés.Toutefois, ce lien n'est pas rigoureusement linéaire, étant donné la large répartition des points.

Au-delà d'une certaine intensité de vent (approximativement 7-8 unités), la surface couverte ne paraît pas s'accroître de manière proportionnelle, ce qui pourrait suggérer un phénomène de saturation ou des éléments restrictifs (tels que la disponibilité du combustible, l'humidité ou la topographie).

Il est également important de souligner que la puissance du vent joue un rôle crucial dans la diffusion des incendies, puisqu'elle transporte de l'oxygène et les particules embrasées amplifient ainsi la vitesse de propagation et l'étendue touchée. Ceci justifie l'orientation initiale à la hausse.

3. Force du vent par zone géographique

Dans cette section on va analyser comment la force du vent est influence par zone geoegraphique sur les incendies

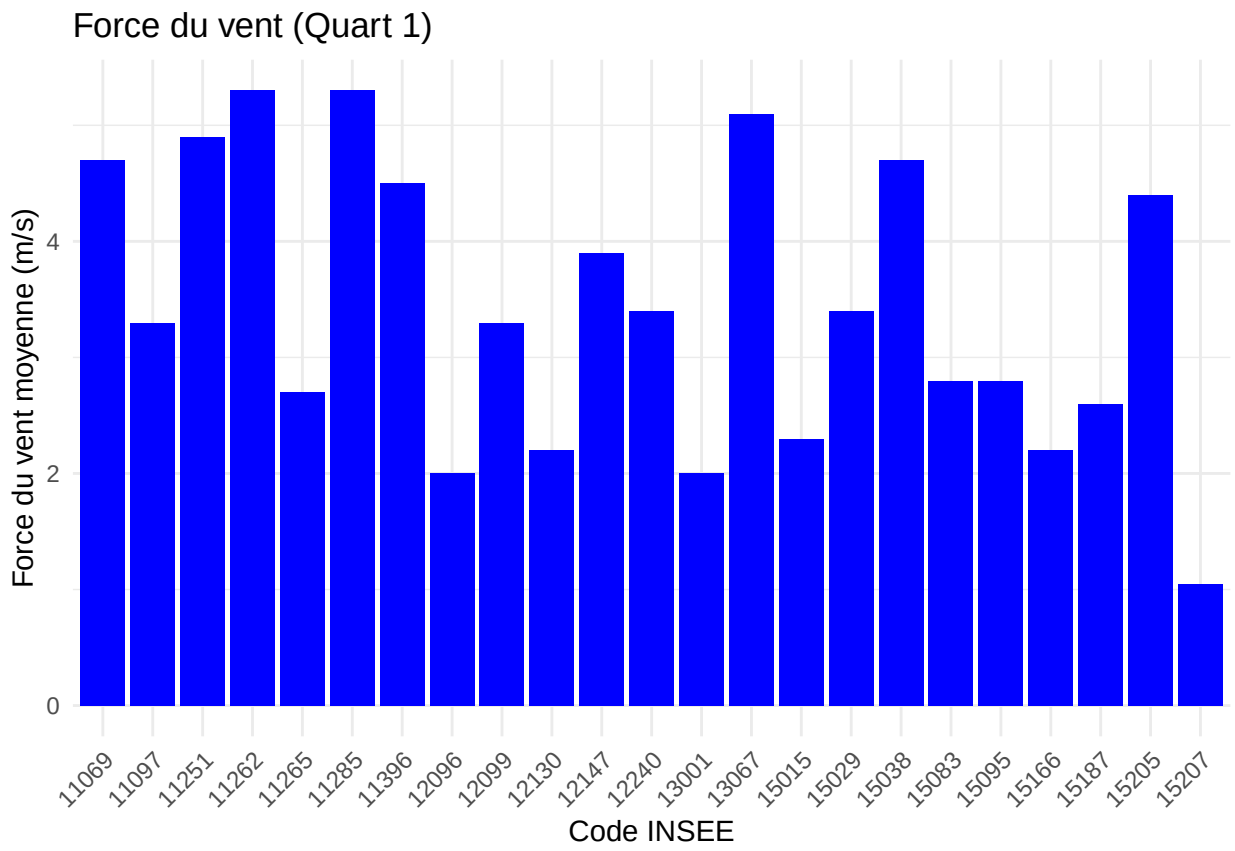
```r
# Charger les librairies
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(gridExtra)  # Pour afficher plusieurs graphiques
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
# Charger les données
data <- read.csv("../Exports/export_vents.csv")

# Sélectionner un sous-ensemble des données (100 premières lignes)
data_sample <- data[1:100, ]

# Vérifier si les colonnes existent
if (!("code_INSEE" %in% colnames(data_sample)) | !("Force_vent_med" %in% colnames(data_sample))) {
  stop("Les colonnes 'code_INSEE' ou 'Force_vent_med' sont absentes du dataset.")
}

# Regrouper et calculer la moyenne de la force du vent
data_summary <- data_sample %>%
  group_by(code_INSEE) %>%
  summarise(Force_vent_moyenne = mean(Force_vent_med, na.rm = TRUE))

# Vérifier que data_summary n'est pas vide
if (nrow(data_summary) == 0) {
  stop("data_summary est vide. Vérifiez vos données.")
}

# Diviser les données en 4 groupes égaux
n <- nrow(data_summary)
quart <- ceiling(n / 4)

data_part1 <- data_summary[1:quart, ]
data_part2 <- data_summary[(quart + 1):(2 * quart), ]
data_part3 <- data_summary[(2 * quart + 1):(3 * quart), ]
data_part4 <- data_summary[(3 * quart + 1):n, ]

# Fonction pour créer un graphique
plot_wind <- function(data, title) {
  ggplot(data, aes(x = factor(code_INSEE), y = Force_vent_moyenne)) +
    geom_bar(stat = "identity", fill = "blue") +
    labs(title = title, x = "Code INSEE", y = "Force du vent moyenne (m/s)") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
}

# Générer les 4 graphiques
p1 <- plot_wind(data_part1, "Force du vent (Quart 1)")
p2 <- plot_wind(data_part2, "Force du vent (Quart 2)")
p3 <- plot_wind(data_part3, "Force du vent (Quart 3)")
p4 <- plot_wind(data_part4, "Force du vent (Quart 4)")

# Afficher les 4 graphiques en 2x2
grid.arrange(p1)
```



1. **Analyse Informatique**

Pour cette sous-question, nous avons adopté une méthode unique consistant à segmenter notre graphe en quatre sous-parties en raison du volume considérable de données, mesuré par le code INSEE.Ainsi, nous procédons à cette division en quatre parties.Nous allons détailler une fois l'analyse informatique du code tandis que les autres seront consacrées aux analyses statistiques.

Dans ce graphique, nous examinons la force moyenne du vent enregistrée pour diverses municipalités, ces dernières étant identifiées par leur code INSEE.

Nous avons utilisé diverses bibliothèques pour réaliser ce graphique, notamment **ggplot2** pour sa conception, **dplyr** pour le traitement des données et **gridextra**afin de présenter plusieurs graphiques dans une seule figure.

Par la suite, nous importons les données à partir d'un fichier CSV nommé vents.csv.

Pour cette analyse, un sous-ensemble des 100 premières lignes a été utilisé, étant donné le volume important de données.

Pour garantir une analyse précise et des données complètes, nous procédons d'abord à un test conditionnel via une instruction 'if' pour les colonnes d'attributs **Code_INSEE** et **Force_vent_med**.Si les données des deux colonnes sont absentes, nous interrompons le script et montrons un message d'erreur.

Par la suite, on procède à l'agrégation des données par le **Code_INSEE** et on calcule la moyenne de la colonne **Force_vent_md**,en omettant les valeurs manquantes.En d'autres termes, nous les regroupons selon le **Code_INSEE**.

Par la suite, si le groupement de notre table ne donne qu'une table vide, nous aurons une erreur.

Comme le volume des données est énorme, nous avons décidé de les segmenter en plusieurs groupes, chacun contenant quatre éléments.

Pour déterminer le nombre de lignes dans le DataFrame lors du regroupement des graphes en quatre catégories, nous avons utilisé la méthode **nrow**.

Par la suite, nous avons divisé le nombre de lignes par 4, puis utilisé la méthode **ceiling** pour arrondir le résultat de cette division vers le haut. Cela nous permettra, sur le plan mathématique, de garantir que si le nombre n'est pas divisible par 4, le dernier groupe comportera moins de lignes, tandis que les groupes précédents auront un nombre identique ou supérieur de lignes.

Il est maintenant temps de les répartir en quatre groupes, ce qui nous amène à définir quatre variables :

1. Pour le groupe initial, nous avons employé **1:quart**, ce qui nous autorise à choisir les lignes de 1 jusqu'à quartsummary.
2. Pour le second groupe, nous avons utilisé **(quart + 1) : (2*quart)**Ainsi, ce groupe inclura les lignes de quart+1 jusqu'à 2*quart.
3. Pour le troisième groupe, nous avons utilisé **(2 * quart + 1) : (3 fois le quart) donc ce groupe inclura les lignes de 2*quart+1 jusqu'à 3*quart.
4. Pour le quatrième groupe, nous avons employé **(3*quart+1):n**, qui englobera les lignes de 3*quart+1 jusqu'à n. Pour indiquer que n n'est pas divisible par 4, ce groupe comprendra moins de lignes, mais inclura toutes les lignes restantes.

Suite à la segmentation des groupes, l'étape suivante consiste à créer les graphiques. Pour ce faire, nous utilisons la méthode **plot_wind()**, qui nous permet de produire des diagrammes en barres illustrant la force moyenne du vent selon le **code_INSEE**.

Nous commençons par la mise en place du graphique où l'axe X représente le Code_INSEE et l'axe Y reflète la force moyenne du vent.

Pour l'ajout des barres, nous utilisons la fonction **geom_bar**. Pour insérer un titre et des étiquettes sur les axes, nous nous basons sur la méthode **labs()**. Comme toujours, nous utilisons **theme_minimal** pour donner un style épuré et en ce qui concerne le graphique, nous inclinons les étiquettes de l'axe des X afin d'améliorer leur lisibilité.

Pour l'affichage des graphiques, nous assignons 4 variables correspondant aux 4 graphiques en utilisant la méthode plot_wind, qui inclut les groupes de données **DataFrame** ainsi que le titre du graphique.

Dans notre cas actuel, pour afficher uniquement le premier graphique, nous utilisons **grid.arrange()**.On place la variable du graphique que l'on souhaite afficher entre parenthèses.

4.3.2.4 Impact des conditions météorologiques extrêmes sur les incendies

4.3.2.5 Effet des radiations solaires sur les incendies

4.3.2.6 Température et nombre d'incendies par période de la journée

4.3.2.7 Relation entre la force du vent et la propagation des incendies

4.3.2.8 Impact de la pression de vapeur sur la vitesse de propagation des incendies

4.3.3 Géographie et environnement

```
import geopandas as gpd
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("../Data/donnees_geo.csv")

chemin_fichier_geojson = "../Data/contour-des-departements.geojson"
gdf_departements = gpd.read_file(chemin_fichier_geojson)

fig, ax = plt.subplots(figsize=(15, 15))

gdf_departements.plot(ax=ax, color="lightgray", edgecolor="black")

ax.scatter(df["longitude"], df["latitude"], color="red", s=50, label="Incendies")


ax.set_xlim([ -5.5, 10 ])
```

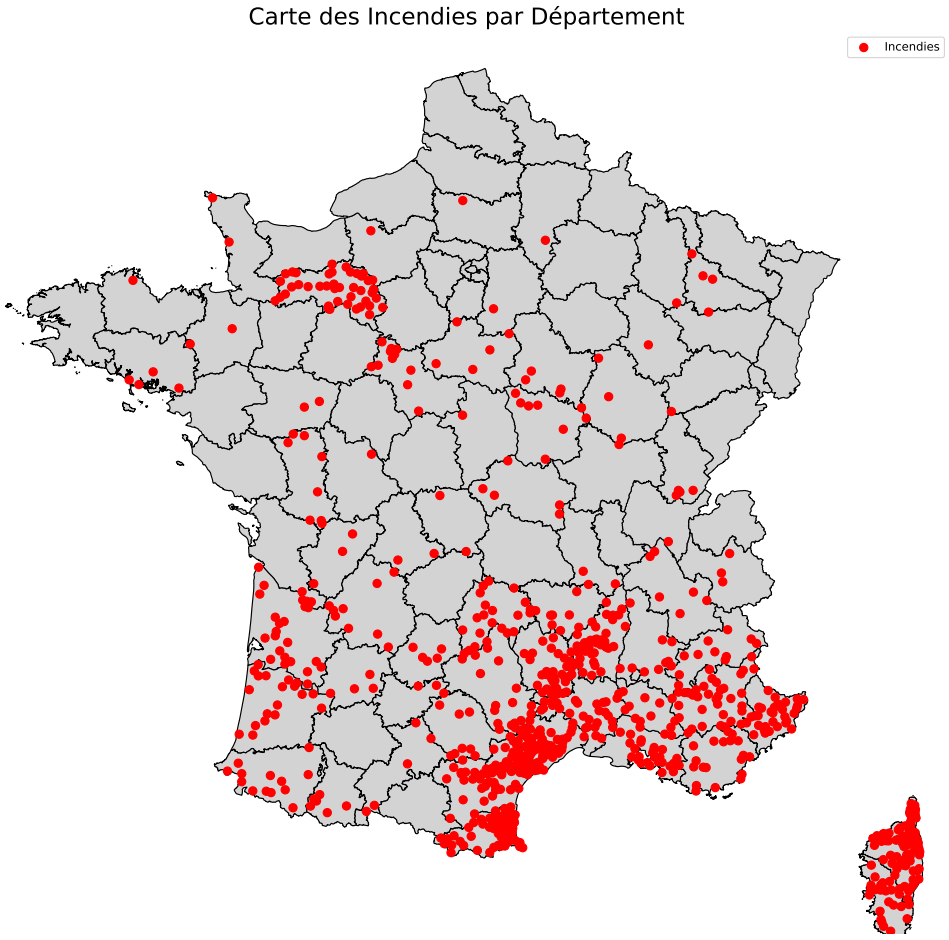
4.3.3.1 Propagation des incendies sur le territoire Francais

```
## (-5.5, 10.0)
```

```
ax.set_ylim([41.5, 51.5])
```

```
## (41.5, 51.5)
```

```
plt.title("Carte des Incendies par Département", fontsize=20)
plt.legend()

ax.set_axis_off()

plt.savefig("carte_incendies_departements_geojson_grande_avec_points_agrandis.png", dpi=600, bbox_inches

plt.show()
```



Nous allons examiner l'analyse de la propagation des incendies à travers le territoire français après avoir utilisé ces informations sur la carte nationale de France.

Décomposons la France, qui est reconnue comme un État composé de régions. Ces régions sont elles-mêmes constituées de sous-régions, qui à leur tour regroupent des départements, et ces derniers, des communes.

En France, on compte 13 régions spécifiquement dans la partie métropolitaine. Nous allons d'abord les nommer, puis commencer leur énumération et enfin procéder à leur analyse.

1. Auvergne-Rhône-Alpes
2. Bourgogne-Franche-Comté
3. Bretagne
4. Centre-Val de Loire
5. Corse
6. Grand Est

7. Hauts-de-France
8. Île-de-France
9. Normandie
10. Nouvelle-Aquitaine
11. Occitanie
12. Pays de la Loire
13. Provence-Alpes-Côte d'Azur

Nous procéderons à l'examen de ces 13 régions individuellement, pour finalement en arriver à une conclusion.

La région **Auvergne Rhône-Alpes** présente un taux d'incendies relativement bas, l'une des raisons étant qu'elle jouit d'un climat plus diversifié que celui d'autres régions du sud.Cette région a des altitudes plus élevées qui entraînent une température plus fraîche et un climat plus humide, ce qui diminue le danger d'incendies de grande envergure.

Le climat tempéré de la région **Bourgogne Franche Comté**, caractérisé par des hivers rigoureux et des étés modérés, explique son taux d'incendies inférieur comparativement à la région Auvergne Rhône-Alpes. Cette région abrite des montagnes et des collines comme le Jura et les Vosges, qui sont habituellement plus humides et moins prédisposées aux incendies forestiers. De plus, cette zone est surtout composée de forêts de feuillus (chênes, hêtres, érables) et de prairies qui présentent une moindre inflammabilité par rapport aux résineux ou aux maquis méditerranéens.

La région de **Bretagne**, en raison de son climat océanique et de ses températures modérées tout au long de l'année, présente un taux réduit d'incendies.Les hivers sont tempérés et les étés restent frais, accompagnés d'une humidité importante. En outre, les précipitations régulières et l'humidité persistante durant toute l'année préservent la végétation plus verdoyante et moins aride, contribuant ainsi à réduire le danger d'incendies.

Il y a effectivement certains incendies dans la région **Centre-Val de Loire**.Cette région jouit d'un climat continental, caractérisé par des hivers rigoureux et des étés plus tempérés en comparaison avec les régions du sud de la France. La région se compose de plaines et de collines, ce qui réduit la probabilité d'incidents majeurs et fait baisser le taux d'incendies dans le centre du Val de Loire.

La Corse est une des zones sensibles aux incendies en raison de divers éléments géographiques, climatiques, etc.La Corse jouit d'un climat méditerranéen marqué par des étés extrêmement chauds et arides, ce qui favorise l'apparition aisée d'incendies.Les températures estivales peuvent atteindre des sommets extrêmes, créant un environnement propice à la diffusion rapide des incendies. Cette zone peut être exposée à des rafales puissantes comme le Mistral qui facilitent la diffusion des feux.Ces courants d'air puissants et généralement secs peuvent propulser les flammes sur de vastes étendues, compliquant ainsi l'éradication des incendies.

La région **Grand-est** présente une moindre susceptibilité aux incidents, mais elle n'est pas totalement épargnée.Cette région jouit d'un climat diversifié qui subit des influences à la fois océaniques et continentales.Les hivers sont rigoureux, les étés sont torrides, cependant ils ne parviennent pas à égaler ceux du sud de la France en termes de températures.L'incidence des périodes de sécheresse est en baisse, diminuant ainsi le danger des incendies. En outre, cette région est dotée de magnifiques zones montagneuses et de vastes plaines. Cette région bénéficie d'une pluviométrie plutôt stable, surtout en automne et en février. Ces précipitations contribuent à conserver un certain degré d'humidité dans le sol et la végétation, ce qui diminue les risques d'incendies.

Le climat humide, la présence de forêts de feuillus, le relief plat et les vents modérés contribuent à rendre la région des **Hauts-de-France** relativement moins vulnérable aux incendies.Ainsi, les conditions météorologiques et géographiques sont généralement moins favorables aux incendies sévères que d'autres zones situées au sud.

L'Île-de-France, étant la région où se trouve la **capitale**, est la plus densément peuplée et urbanisée de France. Elle est moins sujette aux grands incendies que les régions situées au sud du pays. En revanche, cette région n'est pas entièrement protégée contre les incendies. Cette région jouit d'un climat océanique tempéré caractérisé par des hivers rigoureux et des étés doux. En outre, la topographie de l'Ile-de-France

est principalement plane ou légèrement vallonnée, en contraste avec les régions montagneuses telles que les Alpes ou les Pyrénées.

La région de **Normandie** affiche un taux réduit d'incendies.Grace à son climat océanique tempéré, son niveau de précipitations constant, sa topographie relativement plane et sa végétation essentiellement feuillue, la Normandie est moins sujette aux incendies que d'autres zones en France.Ces facteurs contribuent à une augmentation de l'humidité dans l'environnement, réduisant par conséquent le risque d'incendies. Néanmoins, des incendies sporadiques peuvent encore survenir, principalement dans les régions boisées et rurales pendant les périodes de sécheresse ou de chaleur intense, même s'ils restent relativement peu fréquents par rapport aux zones plus chaudes et sèches du pays.

La région **Nouvelle Aquitaine**, située dans le sud-ouest, est susceptible de subir des incendies, notamment à cause de son climat et de sa végétation. Cette région présente un climat plutôt océanique à l'ouest et méditerranéen à l'est, ce qui signifie que les étés peuvent être particulièrement chauds et secs, surtout dans les zones limitrophes de l'Espagne et du Pays Basque. Par ailleurs, les vagues de chaleur peuvent entraîner des phases d'aridité prolongées qui créent un environnement propice aux incendies.

L'Occitanie est une des régions les plus vulnérables aux incendies en raison de divers facteurs géographiques, climatiques et écologiques.Dans sa partie méridionale et sud, cette région jouit d'un climat méditerranéen avec des étés chauds et secs.Durant l'été, les températures peuvent grimper à des sommets élevés et la région est susceptible de connaître des phases d'aridité prolongées, des circonstances propices aux incendies. Dans la propagation des incendies, le vent, notamment le Mistral et la Tramontane qui balaye occasionnellement la zone, a un impact crucial.

La région du **Pays de la Loire** présente une exposition moindre aux incendies par rapport à d'autres régions. Le climat de cette zone est océanique, c'est-à-dire qu'il offre des hivers doux et des étés tempérés, avec des précipitations régulières tout au long de l'année. Les vagues de chaleur se produisent généralement dans d'autres zones, comme le sud-est de la France, ce qui rend la région moins susceptible aux incendies liés à la canicule.

Pour conclure, la dernière région est la **Provence Alpes Côte d'Azur**, qui se trouve dans le sud-est et subit une forte influence de divers facteurs climatiques et géographiques.Cette région jouit d'un climat méditerranéen, marqué par des étés très chauds et secs ainsi que des hivers doux et humides. Durant l'été, les températures peuvent atteindre des niveaux extrêmement élevés avec des vagues de chaleur qui se produisent fréquemment. On peut également observer que cette zone est celle où le taux d'incendies est le plus élevé.

Suite à l'étude de la progression du taux d'incendies sur la carte nationale que nous avons réalisée, il est évident que la région où le taux d'incendies est le plus élevé est la **Côte d'Azur**. En revanche, les zones où le taux d'incendies est le plus faible incluent des régions telles que les **Pays de la Loire**, **Normandie**, etc. Il est donc clair que les conditions météorologiques et la localisation géographique de la ville ou de la région ont un impact crucial, surtout lorsque les conditions climatiques sont particulièrement propices aux incendies. De même, la nature des forêts situées dans les zones urbaines a une incidence significative sur l'expansion des feux.Bien qu'il existe une différence de taux d'incendies entre la ville et la montagne, cette question sera examinée dans le cadre d'une autre problèmatique.

4.3.3.2 Comparaison de la fréquence des incendies dans les régions montagneuses vs basses altitudes

4.3.3.3 Variation de l'altitude par region

4.3.3.4 Distance côtière et risque d'incendie

4.3.4 Urbanisation et activités humaines

4.3.4.1 Comparaison des incendies entre les zones rurales et urbaines

4.3.4.2 Activités humaines à risque (travaux/particuliers)

4.3.4.3 Profil temporel des incendies criminels

4.3.4.4 Facteurs prédictifs des incendies criminels

4.3.4.5 Impact cumulé du climat et de l'urbanisation

4.3.5 Vulnérabilité et analyse de survie

5. Ressources

5.1 Ressources sur la Partie Informative

Jean-Luc CARTAULT, CLAIR, B., & KAPP, D. (2025, January 29). INCENDIES : Le phénomène physique. Encyclopædia Universalis. https://www.universalis.fr/encyclopedie/incendies/2-le-phenomene-physique/

Contributeurs aux projets Wikimedia. (2004, July 23). feu violent et destructeur. Wikipedia.org; Fondation Wikimedia, Inc. https://fr.wikipedia.org/wiki/Incendie

Risque incendie : causes, conséquences et moyens de lutte. (2021). Preventica.com. https://www.preventica.com/magazine/dossiers/prevention-du-risque-incendie-comment-garantir-la-securite-des-personnes-et-des-biens-11032021/risque-incendie-causes-consequences-et-moyens-de-lutte

5.2 Ressources sur la Partie Informatique

Cours et Tutoriels sur le Langage SQL. (2025). SQL. https://sql.sh/

Python Software Foundation. (2019). 3.7.3 Documentation. Python.org. https://docs.python.org/3/

Bien démarrer avec la documentation GitHub - Documentation GitHub. (2025). GitHub Docs. https://docs.github.com/fr/get-started

Qu'est-ce qu'une base de données ? Définition et fonctionnement. (2021, July 26). Hubspot.fr. https://blog.hubspot.fr/marketing/base-de-donnees

5.3 Ressources sur la Partie Statistique

Dérobert, N. (2025). Paramètres statistiques - Position et dispersion. Commentprogresser.com. https://commentprogresser.com/statistique-parametre-statistiques-moyenne-mediane-etendue-ecart-type.html